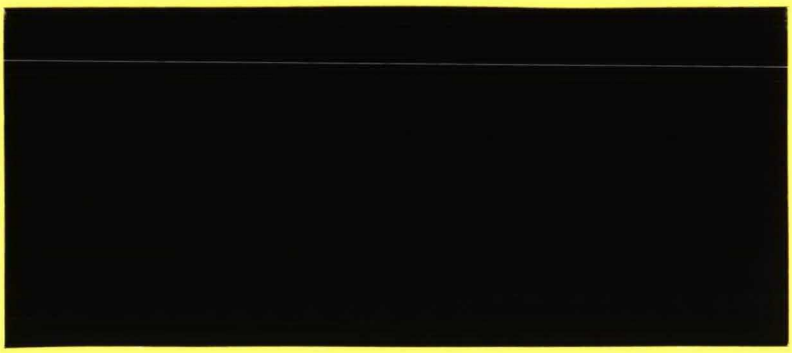
Information retrieval (Part I)

Paijmans, J.J.

Publication date:
1992

Document Version
Publisher's PDF, also known as Version of record



ITK

MEMO

ITK

NSTITUTE FOR LANGUAGE TECHNOLOGY AND ARTIFICIAL INTELLIGENCE INSTITUUT VOOR TAAL- EN KENNISTECHNOLOGIE

ITK Research Memo
january 1992

Information Retrieval
Part 1: Introduction

Hans Paijmans

No. 11

Table of Contents

Part I.

I. Short history of IR systems.

I. Short history of IR systems.

1. This memo.

The aim of this memo is to give a concise inventarization of the vocabulary and techniques used in the discipline of Information Retrieval against the general background of an emerging model of the field. I hope that it will aid students and researchers in computational linguistics and natural language processing in obtaining a better view of this field and that it will clarify the current state of affairs in this discipline, suggest some literature and explain at least a part of the vocabulary. The way this memo is organized will serve as a framework for the first course in information retrieval, to be given in januari 1992; comments and critique are therefore explicitly sollicitated. For this reason it will be printed in two parts, part I: Basics and part II: Document representations, each about fifty pages in length.

In this first part we will start with a short overview of the history of information retrieval and the explanation of some terms in their historical context. This will be followed (chapter II) by an intuitive description of information systems in general and of information retrieval (IR) in particular. The aim of this description is to give a few working definitions of key areas and to create a background against which to proceed, emphasizing the importance of the document representation. As access to the document representations is almost as important as the representations themselves, we will give a short discussion of access methods in a section about databases (chapter III) and an overview of the traditional index-based IR-models. The first part will be concluded by a short chapter IV introducing several kinds of documents and document collections.

In the second part of the memo we will concentrate on several properties of documents as relevant for information retrieval (chapter V), followed by an attempt to sum up the existing document representations. These representations will be grouped in the last two chapters; VI and VII.

2. A short history of IR-systems.

2. 1. The manual era: classification systems.

Although Davies [Davies, 1986, p.264] mentions the ancient greeks in connection with classification systems, it is not clear whether classification systems of any kind were used in the great libraries of antiquity. It is probable though that documents that were felt to belong together, were stored together, be it only by language, author or even physical dimensions. It is also known that books were

4Society

41 Material life of man; physical aspects of living; everyday things

41 A housing
- 1 civic architecture; edifices; dwellings
- 2 interior
- 3 parts
- 4 accessories
- 5 annexes
- 6 garden
- 7 household effects and furniture
- 8 use of the house

41 B the fire, the hearth, lighting
- 1 lighting the fire, kindling
- 2 heating
- 3 lighting, lamps
- 4 afire
- 5 firefighting

41 C Eating and drinking
- 1 nutrition
- 11 eating
- 12 drinking
- 2 kitchen, cellar
- 25 preparing food, cooking
- 26 utensils
- 3 table
- 31 table silver, cutlery
- 32 drinking vessels
- 35 'trionfi di tavola'
- 4 family meal-times
- 5 celebration meal, banquet
- 6 foodstuffs
- 7 drinks, drugs, stimulants
- 9 starvation, famine

41 D fashion, clothing
- 1 fashion
- 2 clothing, costume
- ....

ICONCLASS. This system tried to classify iconographic contents of pictures according to the UDC-principles.

I. 1. Part of the Iconclass system.

Note the difference in emphasis compared to the ICONCLASS system on the last page. The Dewey system is clearly aimed at a user, who wants to know about cooking; ICONCLASS is for a user, who wants to describe a scene.

641.5 Cookery

Preparation of food with and without the use of heat. Observe the following table of precedence, e.g. outdoor cookery for children 641.5622 (not 641.578)

for special situations	641.56
Quantity, institutional, travel, outdoor cookery	641.57
Time-and-money saving cookery	641.55
With specific appliances, utensils, fuels	641.58
for specific meals	641.52 - .54
for specific types of users	641.51
Characteristics of specific geographic and ethnic environments	641.59

Class menus and special planning in 642.1 - 642.5

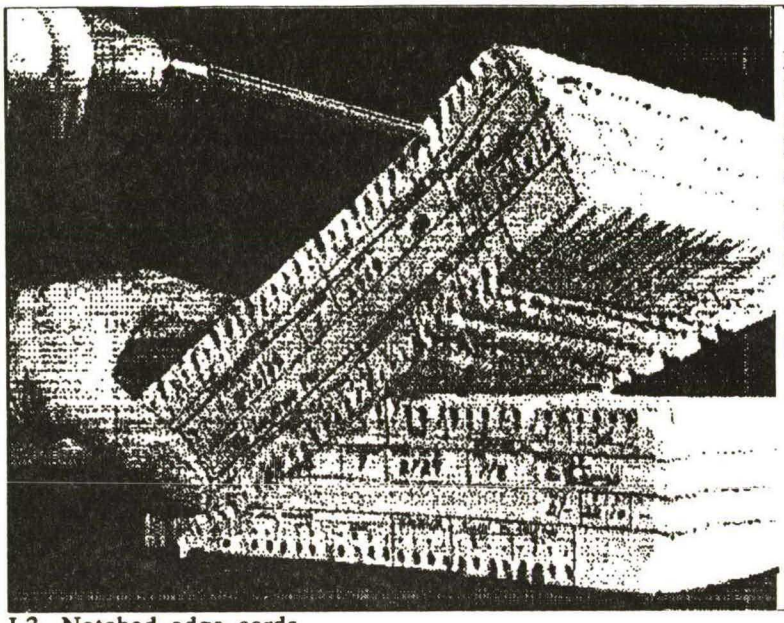
For cookery of and with specific materials see 641.6; specific cookery processes and techniques, 641.71; specific kinds of composite dishes, 641.8

I.2. Dewey's classification system.

lent to other libraries or to private persons, which supposes an administration to keep track of those costly, handwritten volumes[1].

Classification systems as a means to retrieve books in a library did come into their own from late 19th century and the 20th century. These retrieval systems belonged to the group of *assigned indexes* by reason of the fact that first the classification system was conceived and described and the documents were *assigned* to them afterwards by attaching keywords or keyphrases to them. Together with the ubiquitous index on author this subject- or systematical index survives to this day. The choosing of appropriate terms and the syntax of the combinations of the terms often gave birth to quite intricate systems, for which the term *index language* was coined. Examples are Dewey's classification system (fig. I.2) or the Universal Decimal Classification System. See Foskett [Foskett, 1982] for an exhaustive survey of library systems. It is interesting to note that similar classification systems have evolved for non-textual collections. The dutch ICONCLASS-system [VanderWaal, 1955] for example covers the content of pictures i.e. iconograpy (fig. I.1).

1 In the library of the archaelogical institute of Amsterdam University hung a photograph of an inscription found at the site of an Hellenic library (200BC). It translated as follows: "We have sworn to wash our hands before reading the books, not to damage them and to bring them back in time..."



Notched edge cards. Presence or absence of attributes are marked by holes and notches in the edge of the cards. Cards can be separated by inserting a needle and lifting the 'holes'.

[Jolley, 1974].

I.3. Notched edge cards

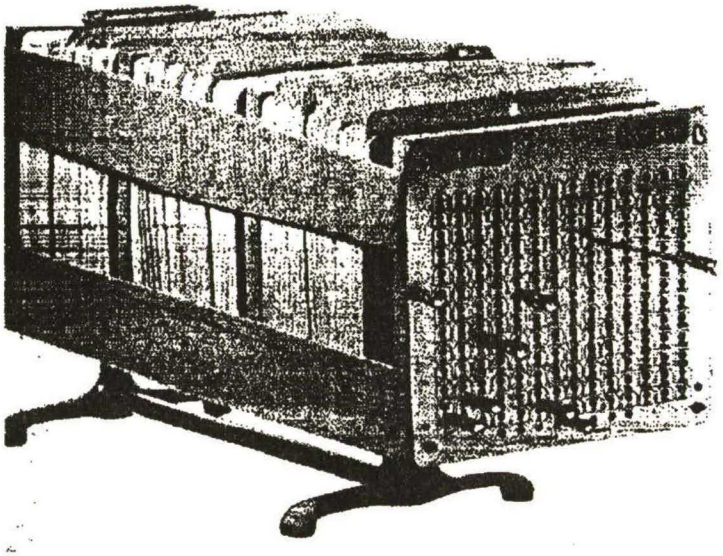
2. 2. The mechanical age: inverted systems.

With the introduction of the computer and automation another approach to indexing became important: derived indexing. *Derived indexing* as opposed to assigned indexing does not try to assign the document under consideration to an existing classification, but on the contrary tries to *extract* from the document those words or phrases, which will subsequently represent the document in the index language. Of course the problem is how to identify those words and phrases in the document, that best describe the contents or 'aboutness' of the document.

The first attempts to 'weigh' words in a document in order to predict their worth as a content-describing keyword date from the late fifties [Luhn, 1958], nevertheless this problem was never really solved. All kinds of probabilistic and heuristic schemes have been proposed, but only a few are adopted by real life systems. The succesful commercial systems like STAIRS work with an inversion of words and documents (for inversion or inverted files see chapter III) without many attempts to weigh the words, although a 'stoplist' of meaningless words (the *function* words in linguistics) is a general feature. Sometimes *stemming* is applied, i.e. suffixes are removed from words. *Lemmatizing* performs a similar function, except that I would like to reserve this word for actions where the *lemma* of the original word is reconstructed rather than the relatively raw trunks that remain after removing suffixes. STAIRS also offered the option to recognize 'paragraphs' in documents, structures very similar to the fields in formatted records, in that they offered field control while searching for keywords.

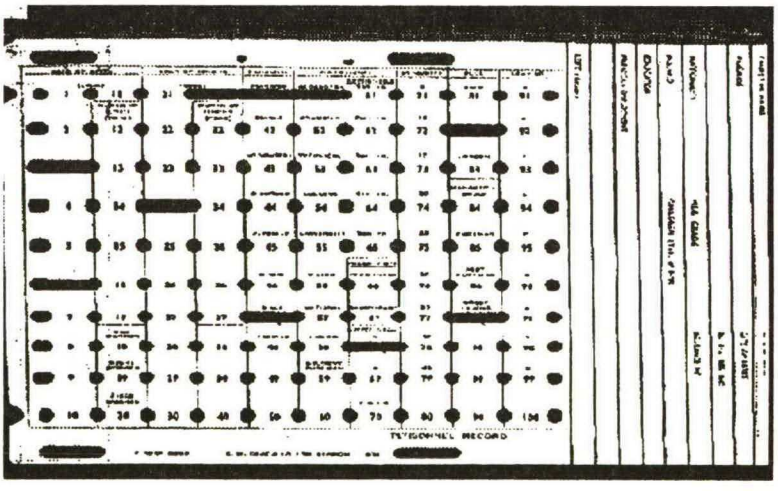
Peek-a-boo card system. Presence of attributes is indicated by slits in the middle of the cards. This increases the number of possible attributes.

If a pin or pins are inserted in the holes that represent attributes to select on, and the cards are subsequently lifted by the pins, the cards, that have slits, will be shifted relative to the other cards and can be lifted out.



Card of a peek-a-boo system. Note the slits. Also note that slits may extend over more holes, making it possible to use non-boolean values.

[Jolley, 1974].



I. 4. Peek-a-boo system (card and cabinet).

Much research has been done, notably by Salton and van Rijsbergen, into strategies to improve recall and precision in such systems like STAIRS, that made documents accessible on occuring keywords. One field of improvement has been the development of several tools and strategies to use at query-time. The combining of keywords by means of boolean operators and adjacency, the thesaurus, fuzzy logic and weighting of individual term-document relationships were all tried; relevance feedback was another attempt to increase recall without sacrificing precision. Also special parts of the documents were singled out for separate indexing and processing (e.g. bibliographies, resulting in the citation index).

> Some of these terms will be familiar to the reader, others might need some clarification. The boolean operators (AND, OR, NOT, XOR) will need no introduction and neither will the relational operators such as EQ, NE, GT, LT etc. Proximity and adjacency operators are such operators as work on the proximity of words (e.g. A SAME B - A and B have to occurr in the same sentence, A ADJ B - A should be adjacent to B). More involved is the concept of fuzzy logic, which tries to introduce elements of probability and uncertainty in logical operations, which is generally implemented in IR by adding weights to keywords and/or operators. Relevance feedback is a technique that, after a search using a query composed of keywords and operators, reports back on the other keywords that are attached to the documents found by that query.

Finally the concepts pre-coordination and post-coordination ought to be mentioned here, two terms that were buzzwords in the documentation community of the seventies and early eighties, although they seem to have lost much of their appeal now.

Foskett observes that there are two kinds of relationship involved in searching: "*...semantic, arising from the need to be able to search for alternate or substitute terms; and syntactic, arising out of the need to be able to search for the intersection of two or more classes defined by terms denoting distinct concepts*" [Foskett, 1982, p.86].

Now if the coordination of the terms is effectuated at indexing time and stored as such in the index language, we speak of *pre-coordinated* indexing. If the terms and concepts are put in the index in a form that enables us to substitute and combine those terms at query-time, we call it *post-coordination*. Pre-coordination comes first, historically speaking. Post-coordination is very much dependent on computers, although "peek-a-boo" systems, notched-edge cards (see illustration), optical coincidence systems and similar devices were popular in the sixties and seventies.

2. 3. Hypertext: the revival of the hierarchical database.

The relational data base system has emerged as the most succesful data retrieval tool. Its properties, with its emphasis on formatted fields, precise attributes and narrowly described domains, made it less appropriate for applications in which text and documents had to be retrieved. Its rival, the hierarchical or network data base, depended very much on (ad-hoc) links between data: it was rapidly going out of

fashion when it was suddenly revived in the hypertext-concept (for a concise survey of this field see [Conklin, 1987], also [Verharen, 1989]). In the hypertext concept links are attached to parts of documents or inside small collections of documents, to gain easy access to relevant material in other parts of the document or database.

Hypertext has become very popular in the text retrieval field, notably in the handling of on-line documentation systems. However, it has certain disadvantages in that big volumes of text are not easily managed and there exists a certain risk of *'getting lost in hyperspace'*: unrestricted browsing may cause disorientation of the user in regards to his original query. Also, adding the links manually is a laborious chore and may cause inconsistencies and uncertainties.

2. 4. The future: knowledge representation.

While van Rijsbergen, Salton and many others were working on what we may now call *orthodox* information retrieval, based on inverted files and keywords, other research was just getting into its stride. This research puts less emphasis on attempts to decide on the most important *words* in the document, but tries to extract and define the contents of the document and *reformulate* it in an independent representation, rather than using isolated keywords. The emphasis, of course, lies on extraction techniques and representations, which are suited for automated processing. We have already touched on this research in the last chapter under the name Document Based Knowledge Systems.

An early attempt was FRUMP [Dejong, 1979]. FRUMP tried to assign news stories to prefabricated *templates*, a technique which might be considered a kind of assigned indexing. However, these templates had slots for expected values (e.g. the strength of an earthquake, or the number of casualties), which contradict the notion of a classification system in favour of a different notion: individual document representation, or rather: the representation of the information in a document.

In the eighties several attempts were made to construct information retrieval systems, which used intricate representations for the meaning or contents of documents in an IR-environment, the most notable of which are TOPIC (i.e. the *german* TOPIC, not the TOPIC marketed by Verity inc. and adopted by a number of libraries) and SCISOR [Rau&Jacobs, 1988]. They base themselves on earlier work on knowledge representation, notably in the world of the AI and text-analysis (Schank, Grosz, Mann, Lehnert). However, even in relatively small domains the overhead and complication of the "world-knowledge" was and is prohibitive. Therefore these experiments have not yet been used in real life systems. We will return to such experiments in chapter VII.

Another exciting development is Parallel Distributed Processing (PDP), also known as neural networks. PDP applications just might be able to solve some problems arising from the necessary vagueness associated with information retrieval, be it full text or otherwise.

An important part of the following chapters will be dedicated to descriptions of possible document representations, the prospects of automatizing the generation of these representations for documents and the subsequent storage strategies. For now we will continue with our general survey of IR-techniques.

II. Information systems and information retrieval.

The discipline of Information Retrieval should be considered to be a part of the science of Artificial Intelligence (AI). The reason is that information retrieval and more general information systems are concerned with the retrieval of data and information (we will use the general term '*info*', when we don't want to differentiate between these two concepts) with the ultimate goal to add to the knowledge of the user.

AI is generally considered to be the science and technology of knowledge. Knowledge is by some researchers defined as "*information that is representing collections of highly structured objects*" (see Daelemans, [Daelemans, 1987]). This fits in with the definitions of Teskey ([Teskey, 1989]). Information in its turn is seen by Teskey as "*structured collections of data, i.e. sets of data, relations between data, etc*" and he defines knowledge as "*models of the world, which can be created or modified by new information*".

The differences between data, information and knowledge may be described as follows:

- data is the result of direct observation of events, i.e. values of attributes of objects;
- information is structured collections of data, i.e. sets of data, relations between data etc, and
- knowledge is models of the world, which can be created or modified by new information. (Teskey p.8)

A lot more can be said about the relations between data, information and knowledge. We will try and develop models and definitions for Information Retrieval (IR) in which these relations are more explicit, but for the moment it should be intuitively clear that for this reason too, IR is firmly associated with artificial intelligence.

1. Information systems.

Information retrieval systems belong to the more general class of *information systems*. This term will be used to refer to systems concerned with the retrieval of data and information. We will differentiate between the following three branches:

1. 1. Data Retrieval.

We speak of Data retrieval when there exists a necessary relation between the formal request and the answer. Its criterion is correctness and for every data item in the database there exists an individual access point: if the salary of Jones is

recorded as $32.000, the occurrence of this quantity at a specific place has unambiguous semantics. The traditional relational and network databases may be considered as data retrieval systems.

There is a limited number of relations between the object that is described and the data, but these relations are very stringent. Therefore they are easily formalized as attributes and tuples, i.e. fields in a record. In such systems the concepts of field and *field control* (the ability of systems to restrict actions to selected fields) are crucial as they control the semantics of the data.

Data retrieval systems return data. It is generally left to the user to make sense of the lists with facts, which are the typical output of the DR system; on the other hand he knows exactly what to expect as the result of a query and how it relates to his information need.

1. 2. Information Retrieval.

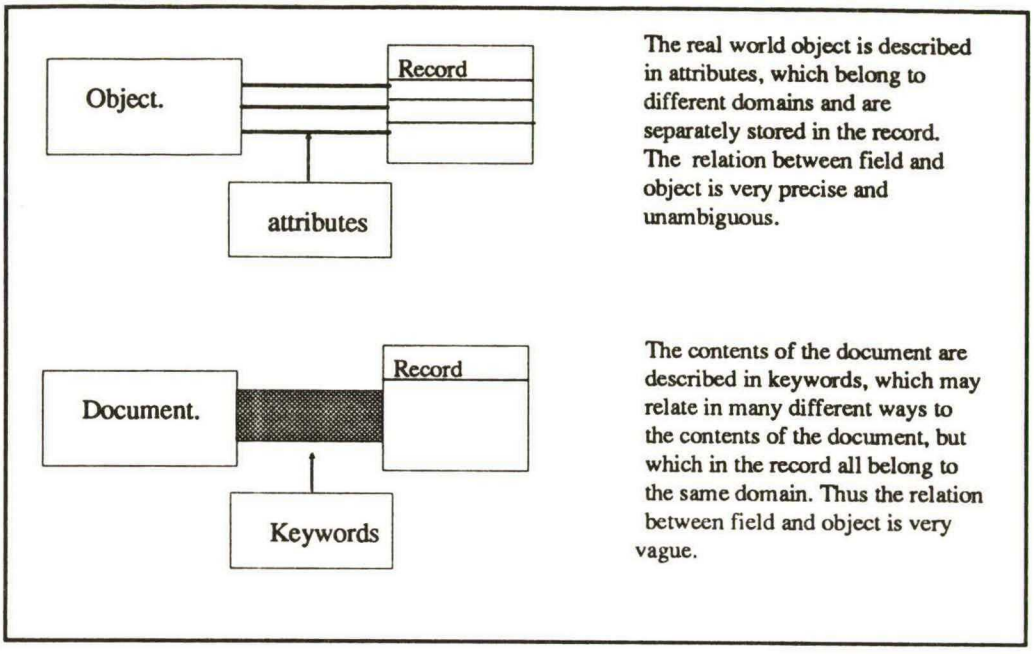
The goals of the users of Information Retrieval Systems are fundamentally different from those of users of 'regular' databases. Although the ultimate goal of consulting an IRS may very well be a piece of data similar to the salary of mr Jones, the user does not search for the data proper, but for documents, which will contain the info he is looking for. In Information retrieval proper there is a *probabilistic* relation between the formal request and the possible answers. Although the formal query may be answered correctly, and *data* may be retrieved successfully, the results may be partly or totally irrelevant to the information need of the user. Therefore the criterion of successful IR is not whether the answers are correct, but whether they are useful for solving the information need of the user.

But if we try to relate questions and answers in such terms, we have to face the reality that the information need of humans, the way they express this need to an IR system and how they relate the answers of the system back to this information need, are very difficult to describe in precise terms.

There has been research in the information needs and subsequent satisfaction of users of IR-systems (notably Online Public Library Catalogues or OPC's (also called OPAC's or Online Public Access Catalogues, see [Sandore,1990] and [Saracevic/Kantor, 1988]), but the designers and researchers of IR systems necessarily work with a very generalized idea of 'the user' and so are tempted to stick to the use of keywords as very general descriptions of the documents.

Now there are a great many possible relations between keywords and the contents of a document (see fig.II.1), and even if it were possible to formalize such relations, the difficulty remains of how to recognize them. Therefore in most IR systems, a small number of properties of the document itself (as opposed to the *contents* of the document) is used as separate attributes (e.g. author or title) and the data items that relate to the contents are formalized as equivalent keywords or possibly as signatures of classification systems.

A traditional IR system is said to contain documents. In reality the relations between the original document and the items existing inside the system are more complicated: we distinguish several disguises or descriptions of documents (document surrogates), which are summed up in more detail in chapter V. The



II.1. Object-record relations in DR and IR.

output of an IR system typically is a list of bibliographic references, although there is a tendency towards the on-line retrieval of the document itself or parts thereof, especially in office automation and in deep documentation systems (see section 3 '*Environments*' below). For this reason Information Retrieval is often called Document Retrieval but some writers maintain a difference between the two. Document retrieval then is considered a specialized form of IR and IR itself shifts toward Question Answering.

1. 3. Question Answering.

A third activity, Question Answering, (QA) may also be defined as an activity, which searches a collection of data or a database with the purpose of retrieving facts and/or information. In Data Retrieval and Information Retrieval systems there is a central object (the record or the document), which acts as the focus of all activity. In QA systems the emphasis is shifted to the user and his information need; the info to be retrieved may be stored in several different structures and be pieced together by the system.

The retrieval of answers often is not retrieval in the sense of directly accessing data items or document descriptions and presenting them to the user, but may well be the result of inferences on the data, the application of rules and intricate user-system interactions. The interfaces for expert systems are often typical QA systems. If data retrieval and information retrieval systems put much emphasis on the user interface and user modelling, they evolve into the direction of QA-systems; QA-systems on their turn are dependent on DR and IR techniques.

> So if I want to know the salary of mr. Jones, I may look it up in a relational database: this is *data retrieval*. If I have no data base system with the salary of mr. Jone and would like to learn more about Jones anyway, I will search for and consult documents, in which his salary possibly would be cited; e.g. in letters about his acceptance of the job or in correspondence between mr. Jones and accountancy: *information retrieval*. If I have a very sophisticated computer system, I might leave it to the computer to guide me into either consulting a database or scan possible correspondence for the facts wanted, or even to infer the salary by comparing Jones' position to similar jobs in the company of which the salary is known: *question answering*.

Another important difference between the three systems is the way in which data and information are stored and accessed. In a *data retrieval system* the data are stored systematically in files such that form and content relate semantically and may directly be manipulated by query-languages that are totally dependent on the ability to limit questions to individual parts of the record (fields), such as SQL.
On the other hand in *Information Retrieval systems* we generally have document surrogates in the form of an *abstract* or even only *bibliographicrecords* in an otherwise perfectly normal relational database system as used in Data Retrieval. The fields of the bibliographic record may be used for field control; the abstract generally is a field containing natural language, it does therefore not have a consistent internal structure and field control may consequently not be applied to this part of the document surrogate. Of course there often exists an index with keywords that occur in the abstracts and that serve as secondary (though not unique) keys to the records.
The data and facts in a *Question Answering system* may be distributed over orthodox databases, rulebases, frames, natural language texts or about every other structure that may be imagined. Although the internal structures are generally opaque to the human user, the QA system is able to communicate with the user, to draw inferences, to model the information need of the user and to influence the direction of the consultation. The emphasis lies on interaction with the user and *user modelling* and less on storage and retrieval issues. For this reason the QA system has many features of a dialogue system. But also, the typical QA system will have to assist the user in the selection of fitting tactics and in keeping control over the direction of his search (for a discussion of information search and control tactics see [Bates, 1979]).

As we will see, this interaction with the user is also growing more important in the IR and DR systems. In the data retrieval systems this is the consequence of the file systems getting more and more complicated; in the information systems the interaction with the user is necessary to define the *information need* of the user in the terms and relations existing in the system. Ultimately the question answering systems will grow to be the front-ends for IR and DR systems, or rather, the IR and DR systems will become modules of general QA-systems.

1. 4. Document based knowledge systems (DBKS).

Although there are other media for storing and transferring information, the single most important vehicle for facts and ideas is the written word. So most of the info a user might want to retrieve from an information system already exists somewhere in a document of one kind or another and even if that info is stored in a different representation, it generally needs some kind of NL-like translation before it can be communicated to the user. Information Retrieval as a means of finding the right documents has been discussed above, but we have stressed the probabilistic nature of this activity and also the fact that the documents themselves have to be retrieved and read, before the info can be used. To assist in this chore, the tendency is to have the documents themselves (or abstracts) on-line and available for inspection after the bibliographic reference is retrieved.

Now if we have the full text of the documents available, it is tempting to try and use more than just selected keywords to retrieve the relevant documents and/or to extract the info wanted by the user from the document. Of course the results of such extraction may play a role in 'normal' Information Retrieval, but at the moment we may witness the first experimental Document Based Knowledge Systems. They will be discussed in the last chapter of part II, chapter VII.

2. Environments.

Another aspect to be considered is the environment in which the information system is used. This environment is responsible for the way the system manifests itself, i.e. for the features which are stressed or omitted, which in its turn are functions of the information stored and of the envisaged users.

When we concentrate on the retrieval of information that is contained in text, we may distinguish several (possibly overlapping) environments:

2. 1. Library systems.

Library systems (and general documentation environments like museums, patent offices etc.) are the traditional stamping ground of the IR worker and most of the following is directly pertinent to this use of IR systems. If not stated differently we will talk about IR in a library environment and more or less synonymous with OPC (but see the aspect *exhaustivity* as described below in 'office systems').

2. 2. Deep documentation.

We consider the term "deep documentation" to mean documentation systems which exhaustively document one subject. Information retrieval is coupled here to the concept of *hypertext*-like navigation and *multimedia*. We will refer to these systems by the abbreviation DDS.

2. 3. Author systems or editorial support systems.

Author systems are combinations of word processor, spelling checker, style checker and retrieval system, all working together to support authors in the writing of text, generally informative text. An author system in itself is not an IR-system, but

there are niches in it for information retrieval or parts thereof: e.g. a thesaurus, fact retrieval, information retrieval proper etcetera.

2. 4. Office automation.

The number of documents in an office automation system (OAS) is generally smaller, but they are at the same time more diverse in form and more dynamic than those in libraries. The information in those documents often contains 'hard' data, which is critical for the relevance of the document in a query as opposed to the very general 'aboutness' which is typical for queries in a library system. Typical examples of such 'hard' data are proper names. As we will see in chapter VI, many IR systems will try to avoid using proper names as keywords, as they in most cases are not useful for dividing documents in topical classes. In office systems data like proper names are considered important attributes of a text. Also texts in a OAS are often structured in a (number of) prescribed format(s), or may be parsed to fill such formats, thus reviving the attributes of the data retrieval system.

There is another aspect in IR that may be demonstrated at the hand of the difference between the OAS and, say, the library system. Users of an OPC will often be satisfied after they have retrieved only a few of the documents that answer a certain query. In office systems (law offices, patent offices) it is often necessary to retrieve ALL of the relevant documents. It should be intuitively clear that this need for exhaustivity has a profound effect on the way that an IR system is used.

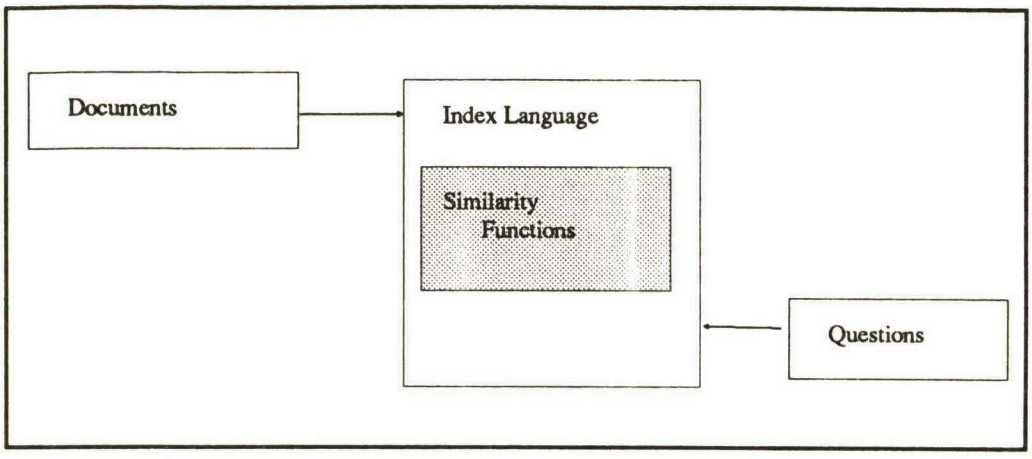
2. 5. Free text data and information storage and retrieval (FTIR).

The availability of (an ASCII representation of) the texts themselves in the database offers the opportunity for more sophisticated analysis of the document and subsequent extraction of data or of indexing information. The documents themselves may be used in any of the other environments mentioned.

In this study we will concentrate on the combination of FTIR and library systems. The distinctions made, however, can help us in obtaining a clear view of the problematic areas and offer pointers to solutions. It should be clear that the individual circumstances and properties of typical texts (e.g. prescribed structures) do have their impact on the information retrieval strategies used, but we will not look very closely to such systems.

3. Information Retrieval: general observations.

Now we have seen how Information Retrieval fits in the more general picture of Information Systems and in what environments IR-like applications may be used. In this section we will try to expose a very general model of IR-systems and the areas in which problems will occur. The terminology that we will use is similar to that used in publications by scientists like Salton, e.g. [Salton&McGill, 1983], which publication may serve as a general reference for the statistical and probabilistic approaches to automated indexing. It should be noted that the use of



II. 2. Information Retrieval model of Salton.

terms like *thesaurus* or *dictionary* in the discipline of IR may be slightly different from that of e.g. linguistics.

We will start with the model, which was drawn by Salton and McGill as the essential IR-system (fig. II.2).
The input left and right consists of:

a. the documents themselves and

b. the queries.

Both undergo a mapping or translation into one (or perhaps several) representation(s) in the

c: indexlanguage.

The solving of the query is effectuated by the application of so-called

d: similarity functions

on the terms in the index language, into which the documents and the queries are translated.

The Index Language (IL) itself may be defined as the sum of these similarity-functions, the translation functions and the indexes; in FTIR-systems the document text itself also becomes a part of the IL. But also the keywords and terms of the IL may be assigned to the document from an existing list of terms, up to the point that such terms do not even occurr in the text. Therefore we use the term *indexing* very loosely in the sense of *all processing aimed at the extraction and representing of information from a document.*

Using this model, it is obvious that the problems in Information Retrieval are concentrated in three areas:

1. in the translation of the documents to the index language, that is: how to create and select one or more representations of the relevant information in the document(s),
2. the mapping of the queries to representations in the index language, which is generally considered to be similar to the mapping of the documents (the validity of this assumption remains to be discussed).

3. the processing and comparing of these representations by means of the similarity-functions to extract the answers to the queries.

Therefore the indexing and query environment is understood best if depicted as a mapping from both query and document onto an intermediate area, defined in the IL. The queries are resolved by applying one or more functions to the translated query and the document representations in the IL: this collection of functions (similarity measures) we will consider also as a part of the index language. However, as the possible similarity-functions are dependent on the representations, for the moment we will ignore these functions and concentrate on the translation of query and document.

3. 1. 'Speaking' the index-language.

What happens in a typical information retrieval action? A user experiences a need for information and takes action to alleviate this need. To do this he will need facts or information, in short 'info'. The first step generally consists of asking other humans for this info; if this does not work, he will turn to the traditional storage of facts, information and knowledge: the written word. What he wants is a document that is 'about' the subject he needs info on and that (he hopes) contains the info he is looking for.

The most common way of retrieving written info is to go to the place in which the user *supposes* that the info needed can be found, i.e. a bookcase, a library or, indeed, a computer. Then he will search the shelves or the catalogue for a title that suggests that the subject of the document is relevant to the information need. Often he will then search the table of contents or the index at the back of the book (the 'BOB-index').

At the very moment he uses the title or the BOB-index to gain access to the info wanted, the user makes a very important assumption: *that he will be able to predict in which words, phrases or expressions the title or index describes the info he wants to retrieve* - and that no other info is described in exactly the same terms. Also, he assumes that the indexer, human or computerized, has used the same terms to describe the books he is looking for. Imagine, for instance, how the biblical story of 'Jonah and the whale' would be described by a biologist, an antropologist, a religious bigot or a freudian psychiatrist.

To coin a phrase: both the user (inquirer) and the indexer will have to 'speak' the index language and the inquirer has to conform to the *Prediction Criterion* as described in chapter III.

In this era of automation you don't go to a bookcase, but to a terminal and the back-of-book (BOB) index consists of a general index with keywords. If you are lucky, the library service you are consulting boasts a thesaurus (a construction, that organizes subjects on their 'aboutness'), but the central problem remains the consensus between the indexing system (human or computerized) and the user about the terminology used to describe the concepts in the documents and their 'aboutness'.

3. 2. Query translation in IR.

Looking at it from the user's point of view, he has to guess how the documents he wants are described in the index language. A two-step translation is involved here: a translation or rather a *realization* of the information need as conceived by the user himself, in terms that the user judges relevant, and a second translation of these terms into the formal semantics and syntax of the index language - i.e. in terms that the system understands. The first formulation we will call the *conceptual query*; the second the *formal query*. To proceed from the first realization of the information need to the query that is acceptable for the IR-system, the user will have to pass the following three stages (for a discussion of research in information gathering see [Rouse&Rouse, 1984]):

1. The user perceives a lack of knowledge and translates this into a information need. This is not as easy as it seems: experiencing a lack of knowledge often implies a lack of terms in which to express this lack of knowledge. The description of the information need will therefore consist of a tentative *circumscription* of the lack of knowledge as his information need (see the 'black hole' in fig. V. 6).
2. He tries to translate this information need into a natural language expression (or in any other suitable way). This is the conceptual query.
3. He then tries to predict which semantics and syntax the system uses to describe the items, mentioned in the conceived query and reformulates the conceived question in expressions the system will accept: the formal query.

Both translations, the proper conceptualization of a query and the translation of a NL-query into the formal query, are studies in their own right. The first translation involves user modelling and the development of search strategies; the second translation (from the conceptual query to the formal query in the indexing language) asks for interfacing techniques and parsers. Efforts to obtain NL query-translation in man-machine interaction may be interesting from several points of view, but there exist any number of alternatives both for the trained client of the system and for the novice - at least for the orthodox keyword-oriented IR systems. In our view it is not even certain if a NL-interface would be the most desirable interface to these IR-systems. This certainly is true for hypertext systems, where natural language navigation really would be prohibitive for normal use.

Also it is often taken for granted that the translation from query to IL will be solved when the problems of the translation from NL document to IL are solved, the implication being that NL query translation is a subset of NL document translation. This is not necessarily true. The environments are very different: the document translation takes place in a controlled environment in which a number of domains are predefined (the subjects of the document collection) and some knowledge or meta-knowledge may accompany the document at translation time. Generally very great quantities of text are involved, which makes statistical techniques feasible and knowledgeable personell may assist the translation process. On the other hand, the query as put by the user initially may well be outside the scope of predefined models and domains, even if the answer may ultimately be

1. Causal antecedent (*What caused him to become angry*?).
2. Goal orientation (*Why did Reagan cut the budget*?).
3. Enablement (*What enabled the depression to occur*?).
4. Causal consequent (*What are the consequences of the budget cut*?).
5. Verification (*Did Reagan increase the military budget*?).
6. Disjunctive (*Is this flower red or blue*?).
7. Instrument/procedural (*How did the people survive*?).
8. Conceptual completion (*Who shot Reagan*?).
9. Expectational (*Why didn't he go to the party*?).
10. Judgmental (*What do you think about Reagan*?).
11. Quantification (*How much money are you in debt*?).
12. Feature specification (*What does Reagan's ranch look like*?).
13. Requests (*Why don't you write your friend a letter*?).

II.3. Lehnert's classification of questions.

found there. Research by Small/Weldon and Schneidermann has shown that users when asking questions in a natural language put significantly more 'invalid' questions. On the other hand the formal query languages were easily learnt, even by people without experience with computers. See also [Baars/Schotel, 1988] for a short discussion and more literature.

As we will see later, it is possible that the querying of intricate knowledge representations will again make natural language necessary. This activity will be located at the user-modelling part of the system and will be aimed at assisting the user in externalizing his information need, rather than translating this information need in expressions for the system.

3. 3. Query translation in DBKS.

In Information Retrieval we will consider a document understood, when those attributes of the document that the prototypical user is interested in, are made explicit and ordered in such a way that they may act as access points to the original documents. For Document Based Knowledge Systems we will extend this condition to the point that the system should be able to answer questions about the contents of the document and to create an abstract of the document, if not in natural language form, then at least in some other form that may be stored and queried by the user.

This brings us to the question what, indeed, may be considered the answering of questions. We will limit ourselves to a short description of questioning- answering as seen from the conceptual dependency point of view.

Graesser and Murachver [Graesser/Murachver, 1985] mention five processing components in the answering of a question:

Once there was a Czar who had three lovely daughters. One day, the daughters went walking in the woods. They were enjoying themselves so much that they forgot the time and stayed too long. A dragon kidnapped the daughters. As they were being dragged off they called for help. Three heroes heard the cries and set off to rescue the daughters. The heroes came and fought the dragon and rescued the maidens. Then the heroes returned the daughters to their palace. When the Czar heard of the rescue he rewarded the heroes.

II.4. Story 1.

1. Interpret the question.
2. Select the appropriate question category.
3. Apply the selected question-answering procedure to relevant knowledge structures.
4. Articulate answers to the question.
5. Evaluate the pragmatic goals of the speech participants.

The interpretation of the question is seen as the reduction of the question to three elements: the *question function*, a *statement element* and a *knowledge structure element*. The selection of the question category or categories is an assignment of the question-type to one of Lehnert's categories (fig.II.3).

Now the available schemes (any of the generic knowledge structures such as frames, scripts, stereotypes etc.) are matched with the question categories.

This translates the question to one or more formulas like:

```
WHY(<man  carries  stick>  <OLD_AGE  scheme>)
```

or

```
WHY(<man  carries  stick><DOG_PUNISHING  scheme>).
```

The nodes in each of the candidate schemes are traced for fitting causal (temporal etc.) nodes and these are checked for constraints.

One of the problems seems to be that for any one piece of text there are many possible statements and inferences and thus questions. Graesser and Murachver report a total of 427 statement nodes (number was manually arrived at) for the story of fig. II.4. This big number seems to prohibit attempts to analyse all possible statements in a document base in advance. Also, and more important, a selection will have to be made of exactly those statements that are of importance for answering such questions as are predicted for typical use.

Increasing association		→		
Increasing clarity of perception		Cognition (awareness)	Memory (temporary	Evaluation (fied memory)
↓	Recognition (concurrent)	Concurrence /θ	Self-activity /*	Association /;
	Convergent thinking (Not distinct)	Equivalence /=	Dimensional (time, space, state) /+	Appurtenance /(
	Divergent thinking (Distinct)	Distinctness /)	Reaction /-	Functional dependence (causation) /:

II. 5. Farradane's operators.

3. 4. The problem of document translation.

The most daunting problem in information retrieval is the translation of the original documents to representations in the indexing language.

Traditionally (manual) indexing languages have two parts, a semantic and a syntactic part. The semantic part consists of a more or less controlled dictionary with keywords, often extended to or accompagnied by a thesaurus. This part may evolve to a complete classification system. The second part is a set of rules that governs the possible combinations of these keywords, often accompagnied by a set of operators (ill. II. 5 and 6). Foskett gives an extensive description of indexing and abstracting in libraries (see [Foskett, 1982]. The task of the human indexer then is to translate the aboutness of the document in the terms of this indexing language and the computerized indexer should assist the human indexer up to the point of doing the same job or a very similar one on his own.

However, we should keep open the possibility that computerized indexing systems may ultimately end up doing very different things. Of course, there exists a strong tradition for users to formulate their information need in terms of the 'human' systems they have become used to in the last few hundred years. Prolonged use of automatized systems may have the effect that users will change the conceptualizing of questions to forms that offer better results on automatized systems.

But for now a small army of human library workers all over the world (or perhaps not so small an army) has been engaged for almost a century in reading books and articles, trying to apply indexing terms from the most exotic systems and storing and cross-referencing those terms for millions and millions of books in many hundreds of libraries all over the world. Many more prospective users have approached these retrieval systems to try and find literature relevant to their information needs and while a scholar may become proficient in the quirks of one or two such systems, he or she may be foundering trying to access the information

/θ Concurrence
1. Mental juxtaposition of two concepts
2. Bibliographical form.

/; Association
1. Unspecified
2. Agent
3. Abstract, indirect or calculated properties
4. Part or potential process.
5. Thing/Application
6. Discipline (subject study).
7. 'Dependent on...'

/* Self-activity
1. Intransitive verb.
2. Dative case
3. 'Through...'

/= Equivalence
1. Synonyms, quasi-synonyms
2. Use

/+ Dimensional
1. position in time and space
2. Temporary state
3. Temporary or variable properties

/(Appurtenance
1. Whole part
2. Genus-species
3. Physical or intrinsic properties

/) Distinctness
1. awareness of a difference
2. Substitutions or imitations

II. 6. Farradanes operators and their applications.

in a third one (this last observation might be understood as one of the arguments in favour of NL query translation).

Of course these observations are not new. The studies of Cleverdon, Lancaster, Salton and many others all point to the conclusion that:

> if two people [...] construct a thesaurus in a given subject area, only 60% of the terms may be common to both thesauruses;

> if two experienced indexers index a given document using a given thesaurus, only 30% of the index terms may be common to the two sets of terms;
>
> if two search intermediaries search the same question on the same database on the same host, only 40% of the output may be common to both searches.
>
> if two scientists [...] are asked to judge the relevance of a given set of documents, the area of agreement may not exceed 60%.

[Cleverdon, 1984].

Cleverdon goes on saying that "*(These) problems [...] may be overcome by [...] using as the input, an extract such as the title and abstract in natural language...*", but he forgets to mention who will generate the extracts and the abstracts and according to which rules. Also noteworthy is the confusion between extract and abstract in Cleverdons text. We will explore the very real differences between abstract and extract later.

Now if human indexers are inconsistent or even erratic, they are at least able to read and understand a document, if only at the semantic level. On the other hand, a computer conceivably would be able to maintain a high level of consistency, but reading and 'understanding' a document by computer poses many problems, not the least being what 'understanding' a document really means. So if we compare Cleverdon's findings with the Blair-Maron study of an automatized IR-system [Blair/Maron, 1985], we may find the same black picture.

In Information Retrieval we will consider a document understood, when those attributes of the document that the prototypical user is interested in, are made explicit and ordered in such a way that they may act as access points to the original documents. For Document Based Knowledge Systems we will extend this condition to the point that the system should be able to answer questions about the contents of the document and to create an abstract of the document, if not in natural language form, then at least in some other form that may be stored and queried by the user.

The set of keywords, which are extracted from a document, or the index terms allotted to it, can be considered as models of the document from the viewpoint of the index language. The vector models as described by van Rijsbergen, Fox and Salton are other ways of document representation; the work of Lebowitz, Rau, Schank and many others again point to alternative ways of modeling documents. What is needed is an exhaustive summarization of all the levels on which a document may be 'understood' that is 'described', or rather 'models' of the document. Alhough in this memo we do not aim at exhaustivity, we will try and give a reasonably complete survey of the state of the art.

III. Databases and Early IR-systems.

A computerized information retrieval system generally is centered around a collection of computerfiles, which is called a database (or data base[1]) or even around several databases. The contents and organization of these databases are responsible for many of the possibilities and for the performance of the IR system. In this section we will consider a number of access methods for documents.

The word 'database' has come to mean a variety of things, especially in the context of knowledge representation and expert systems. Often it is not clear which meaning is used (or even whether one should write 'data base' or 'database'). Therefore it is perhaps necessary to devote a few general words to databases and its peers.

1. Regular databases.

The activity of working with datafiles has grown into a discipline, which has become one of the most important fields in computer science: *data base management.* In the beginning of the sixties the need was recognized for a standardized approach to the use of data files in computer systems and to progressively try and hide the details of storage and implementation, both from end-users and application programmers. This caused the formation of the CODASYL committees and the Data Base Task Group, among other attempts to grasp the problems of data models and standards for data base management in computerlanguages.

But perhaps the single most important step was the recognition by Codd that datafiles could be described by the relational model [Codd, 1970] and the subsequent interest in relational database management systems has sometimes hidden the fact that different approaches of data bases did and do exist (e.g. the aforementioned CODASYL group, which promotes the hierarchical or network model). In information retrieval circles however, the hierarchical model was not forgotten [MacLeod, 1987] and of course, the hypertext-structure [Conklin, 1987] is a network.

---

1 Database (written as a single word) seems to be favoured by the followers of Codd and Date and has become popular in the wake of the relational database. The usance of writing it as two separate words is found by the partizans of the network data base, notably Olle. I will use database, although this does not indicate any special feelings in this respect.

The differences between the relational and the network database (RDBMS and NDBMS) will be familiar to the reader: the relational database organizes its data in tables (relations) and relies on identical fields in the tables to link the tables; the network database views related data as sets and uses explicit pointers to establish the links. For every record there is at least one *primary* key, which identifies that record *uniquely*, and zero or more *secondary* keys, that also may be used to retrieve that record (and possibly more records).

A database is a collection of one, but preferably more (data-)files. Its main functions are threefold [Smith&Barnes, 1987]:

- Mapping between application programs and the logical database by means of functional databases (which may appear as combinations of the data in the logical database).
- Mapping between application programs and details of physical storage.
- Avoiding anomalies, redundancy etc. between the datafiles, which exist in the database.

Another formulation is "*A database (...) is a repository for stored data. In general it is both integrated and shared. By 'integrated' we mean that the database may be thought of as a unification of several otherwise distinct datafiles, with any redundancy (...) partly or wholly eliminated. (...) By 'shared' we mean that individual pieces of data may be shared among several different users..*".[Date, 1981. p.4-5].

"*The essential difference between a data base and a file should be that the former contains cross referencing from one part of the data base to the other... ...it is proposed to define a data base as a cross referenced collection of data records of different types and a file as a collection of records, which are not cross referenced and in which the records are generally all of the same type.*" [Olle, 1980, p.8].

We will also mention the *distributed* database, which may follow the principles of network or relational databases (or indeed any other way of organizing the records), but which distinguishes itself from 'normal' databases in that the functional parts of the system are distributed over different localities.

1. 1. A data base is not just a collection of data.

In many AI and IR textbooks any collection of data, regardless of structure and format, may be called a database, even if it resides in just one file or even in core). This, I think, is not the proper use of the word: if we speak of a database as a collection of more or less similar records in one single file, without cross-references, that should, according to the definitions of Olle and Date, rightly be called a datafile. If the data exist in core, it should be called a (collection of) datastructures, (presumably) filled with data. If a generic name is wanted, the expression *knowledge base* is a better term.

Nevertheless this rather imprecise use of the word has become common in books about AI in general and expert systems in particular. Central to the definition of a database are the concepts of mapping and abstracting from the details of access and storage. The august disregard for the niceties of data base management that is displayed by the AI and expert systems community might be regarded as exactly that: abstracting from details. And of course the fact that the highly formalized relational database has become the prototype of all databases, does not alter the

fact that another approach might be, if not as effective, at least not in contradiction with the original definitions.

1. 2. Document data as database Attributes.

In normal (relational) databases the information about objects usually is stored in records, that are formatted in one way or another. This formatting is realized by matching attributes of the original object with *fields* in the record, fields that generally have a fixed order and length. Relevant properties of the items are translated in atomic values, that fit the domains of these fields. The ability to select individual fields for query and display (called *field control*) is perhaps the most important single tool in data base management.

In all but the most primitive systems, the data related to one single object are organized not in a straight 1 - 1 organization of a record and the object itself, but we see that attributes tend to become objects in their own right and subsequently are organized in new, separate files. So if an employee is an object or an entity in a RDBMS, he gets assigned a record in a file and the department he is working in, becomes an attribute. But as departments are entities in their own right, they tend to split off from the table 'employee' and get organized separately in the table 'department' for normalization purposes.

Now the manner that the data of typical data retrieval systems are collected, is not of importance for the system itself. The situation for typical IR systems is totally different: the manner in which the keywords- and phrases that act as data in an IR system are arrived at are of crucial importance for the functioning of the IR system.

The problem with documents is, as we will see in chapter V, *Document Properties*, that like some of the better equipped mythological monsters, they have three different heads.

1. The document may be seen as an object to be collected and managed: this calls for a precise registration and identification.
2. But it also is a physical object, the properties of which can be measured, weighted and counted: translated to the view of a document as a collection of characters, these characters may be counted and organized. Because of the relatively easy and unequivocal way that these properties can be decided upon and described, we can classify them under the *data properties* as mentioned above. The recognition, storing and retrieving of these data properties as attributes cannot any more be considered a problem. Other attributes, e.g. hierarchical TOC's, are not so easily stored in the formatted fields of a relational system, but the recognition in the document, especially if edited by a modern wordprocessor, does not pose any special problems.
3. But if we try to store the *knowledge* contained in the document in this attribute model (we'll call it 'indexing' for now, although it may be totally different from a back-of-book index), we will see that the relational model immediately breaks down for all but the simplest keyword models. There are a number of very good reasons for this:

1. 3. The Prediction-problem

The format of a record in a database is dependent on the object to be described. To design a format, it is necessary to decide on the properties of the object to be entered and to predict which properties will be asked for (prediction here not to be confused with the use of the word in the term *Prediction Criterion* as mentioned in chapter II).

It is next to impossible to predict the contents of a document in a non-trivial retrieval system and reflect the possible contents in all but the most general domains and attributes. For this reason a relational data base system is all but ruled out.

1. 4. The Consistency-problem

Then there is the problem of getting indexers (or indexing system) and user to agree about the terms that should be used for presenting the info in the document. Using computers for indexing at least adds consistency to the representations of the system, but unless we let computers do the consulting of the database too, we are stuck with those messy humans, who delight in calling a spade anything but a spade. Therefore the orthodox database models, that are very dependent on the exactitude of their data, will not perform satisfactorily.

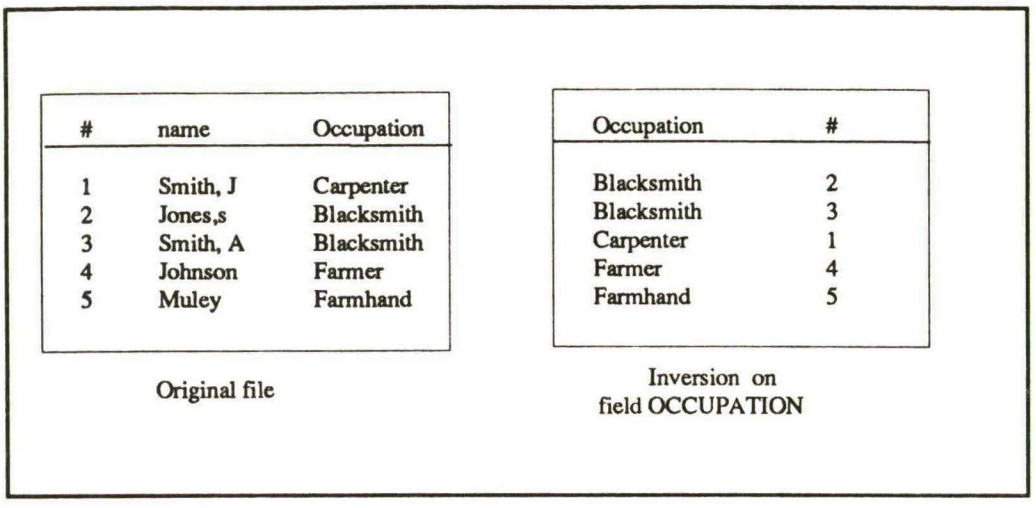
1. 5. The precision/recall-problem.

Directly related to this inherent fuzziness is the fact that most questions put to a IR system will either produce a great number of irrelevant questions or omit possibly relevant answers. Although this has no direct bearing on the organization of the database itself, it certainly has consequences for the interfacing to it. We will cover the precision/recall and similar problems in the section about measurements.

1. 6. The topicality problem.

The essential problem remains how reliably to extract the topicality or 'aboutness' of a document. Again, in itself this is not a database problem and we will return to it with a vengeance later in this memo. But we mention it here for completeness.

The question comes to mind whether (the contents of) NL documents are extraordinary objects, too complex to describe in an orthodox database. The answer obviously is 'yes'. Although there are many objects of comparable or even greater complexity that are described, stored and retrieved satisfactorily in relational and other databases, the contents of NL documents defy all attempts to catch them in the tables of a normal DBMS in other than the most crude representations. Nevertheless files and databases are obviously necessary for any computerized information system and we will now go into the uses they may have.



III.1. Inversion of a file.

2. Database access.

The files in a document based IR system have the following purposes:

1. storage of the document surrogates: the documents as presented to the system (possibly erased after processing).
2. storage of the on-line documents: the documents that serve as final output of the system (possibly only bibliographic descriptions).
3. storage of the document representations: the internal representations extracted from the document surrogates. These representatons function as secondary keys to the on-line documents.
4. storage of general knowledge: thesaurus and other general knowledge representations.

(for a discussion of document surrogates, on-line documents etc. see chapter IV and V).

The essential function of a database is the access function. Therefore we will give a short and general description of the access techniques applicable to text retrieval: *full text scanning*, *inversion*, *multiattribute retrieval* methods and *cluster-based access* methods.

2. 1. Full text scanning.

Full text scanning is a straightforward way of searching documents, which contain strings, that to the user are indicators that the document is important to him (may alleviate his information need). These strings may either be literals or *regular expressions*[2]. The database is read sequentially either to the point that the first (n-th) occurrence of the string is found or to the end. This action has to be repeated for every query; therefore this method is very time consuming.

The advantages are that no overhead in storage is needed other than the document (document surrogates), further that minimal effort is needed on insertions and updates and that the search string to be matched may be of any reasonable length: i.e. the search does not have to limit itself to keywords or keyphrases. Interesting developments of the last ten years have been the use of dedicated hardware for full text scanning (see [Faloutsos, 1985]) and connectionism, holding out the promise of greater fault tolerance and machine learning ([Waltz, 1987]). This falls outside the scope of the present publication.

2. 2. Inversion.

The *inverted file*, that is a regular adjunct of FTIR systems, deserves some attention. In files that consist of tables, the term 'inversion' indicates that a new file is created, in which the record-field order is inverted for one or more fields (see fig. III. 1). If in the original file the access point was the record, which consisted of a list of fields, after inversion the field becomes the access point, where the record is found. In IR-usage this concept sometimes degenerates to mean a list of keywords with pointers to the documents in which they occur. An occurrence in such a list is sometimes called a *posting* and the inverted file may be called an index, a concordance or a dictionary, according to the different authors. We will prefer the term dictionary as an index may have more meanings in IR usance and a concordance definitely is a different concept (if no confusion is possible, we will sometimes use the word index as a synonym for dictionary, because it is generally accepted in IR literature).

A system, based on keywords extracted from uncontrolled NL, has severe limitations, including the following:

1. The synonym problem similar concepts are named differently).
2. The homonym problem (identical words have different meanings).
3. Generic search is difficult, if not impossible.

This makes it necessary that the inverted file expands to include phrases, e.g. 'aluminum welding' or 'fragmentation ammunition' and that relations between keywords and phrases are defined in a thesaurus. Either technique really needs NL understanding, although combined syntactic-statistical methods for phrase-indexing are reported to be succesfull ([Evans, 1991]).

Although inverted systems that were generated automatically, were generally considered as reliable as manually generated indexes, or even better, Blair and Maron in a much-cited article [Blair&Maron, 1985] stated that nevertheless the recall ratio remained far below the expected. In an experiment, aimed at retrieval of 80% of the relevant articles in a STAIRS[3] document database, it was found that in reality only 20% of the relevant documents were retrieved. Worse yet: the users

---

2 Regular expressions are expressions with wich variations on a string or strings may be formulated.

3 STAIRS: STorage And Information Retrieval System; the inverted-file based full text retrieval system of IBM (1972).

were not aware of this fact. So even if automatic indexing such as in the STAIRS system performs better than manual indexing (as Salton maintained in an reaction on Blair and Maron [Salton, 1986], there is ample scope for improvement.

Because of the ease with which inverted files may be genereated, the rapid access to the documents and the ease with which boolean algebra may be applied, they have become very popular in information retrieval systems. As a disadvantage may be mentioned the overhead of the index (50-200% depending on the information in each entry and compression techniques). Also the cost of updating in a dynamic environment with many insertions and updates may become very high or even prohibitive.

The keywords in the inverted file may consist of all words in the original documents. In that case recall and precision compare with the recall and precision of the full text scan (apart from strings that cover more words and searches using regular expressions). If a stoplist is applied, the situation does change. If the keywords in the inverted list are filtered by a *stoplist*,the result of stemming or of a weighting technique, performance will change drastically, according to the techniques used.

2. 3. Multiattribute techniques.

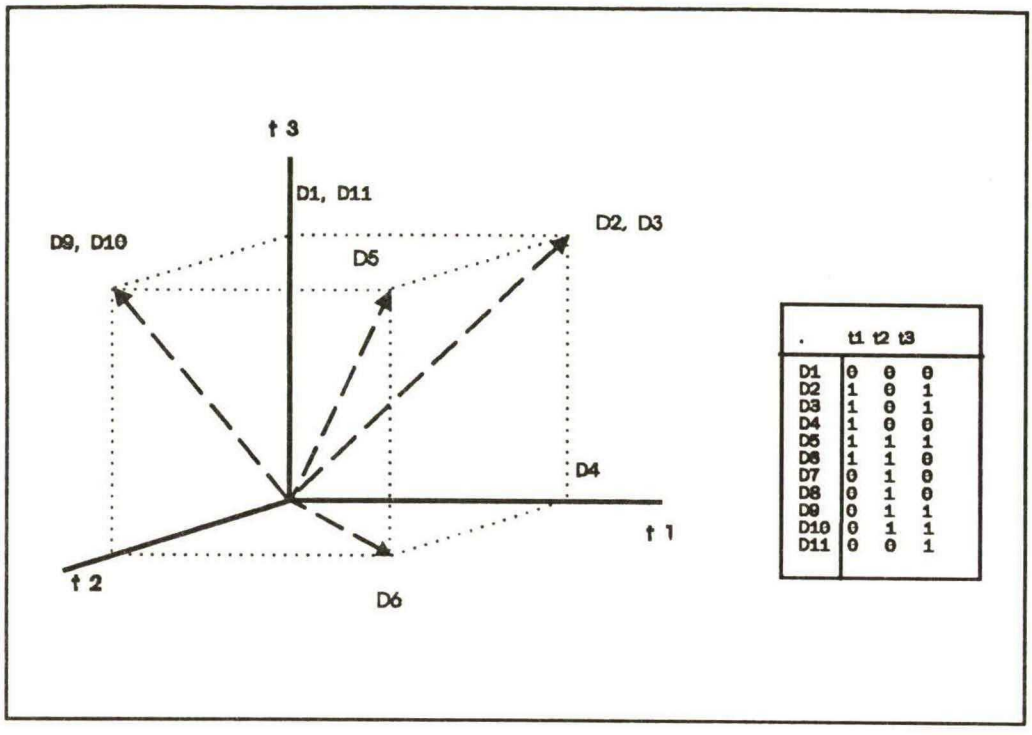
Inversion is not the only way to create secondary access for records. There exists a school in Information Retrieval, which depends very much on bitmaps for storage and retrieval of relevant documents. The *superimposed coding* and *signature* methods hash keywords or n-grams of the contents of the record to bitpatterns (the signatures). These bitpatterns are either concatenated or superimposed (OR-ed) to represent the document. These techniques are reported to be efficient in relatively small, dynamic databases; response time suffers though when the database grows.

The problem with these methods is that there are typically many keywords in a textual database but that there are many zero's for every individual document. On the other hand the postings in the signature effectively identify the document and could be used in a hashing algorithm to access the document (*multiattribute hashing*). The big number of keywords, the fact that this number changes dynamically with updates and the great number of zero entries create practical problems. However, for relatively small databases multiattribute methods have been used with success for secondary key retrieval.

These systems concentrate on rapid storage and retrieval of the information and offer not much scope for improvement in the field of representation. For this reason we will only mention them here in the section about databases and will not enlarge on them. See Faloutsos and Christodoulakos for more information [Faloutsos, 1986]; [Faloutsos&Christodoulakis, 1984]; also [Chudacek, 1983].

2. 4. Clustering.

Of course, if we could devise a way of bunching similar documents together, this also would improve efficiency and as an added bonus would enhance the representation of the contents of the database. But how does one decide on the similarity of documents? One of the answers is clustering.



.	t1	t2	t3
D1	0	0	0
D2	1	0	1
D3	1	0	1
D4	1	0	0
D5	1	1	1
D6	1	1	0
D7	0	1	0
D8	0	1	0
D9	0	1	1
D10	0	1	1
D11	0	0	1

III.2. Document vectors.

A way to represent a document by its keywords is based on vectors. Let t be the number of distinct words (keywords) in a textual database. Now each document may be presented as a vector in t-dimensional space that has a keyword for every axis: a zero signifying that the keyword does not occur in the document and a non-zero (a one for binary, a number between 0 and 1 for weighted entries) indicating a posting.

The weighting of the terms is an interesting subject in itself. One of the reportedly most effective methods is the *tf-idf* weigthing, in which each term is represented by the product of its term frequency (number of occurrences in the document) and a function of its inverse document frequency (total number of occurrences in the document). The similarity or difference between the individual documents may now be measured in a number of ways, e.g. by comparing the angle between document-vectors (the cosine measure) or the euclidean space between the ends (see fig. III. 2). See chapter VI: *Document Representations,* for a discussion.

3. A short survey of Retrieval Tools.

An important consideration, in respect to both the general design of the system and the ultimate user satistaction, is the strategy and tools that are to be used at retrieval time. One might think that they are closely bound to the docrep, as the docrep is the object they operate on. But as we will see there exists a rather general collection of tools, that may be applied to almost every collection of docreps of almost every system. Advanced systems may add new tools to cater for

their special features, but the boolean and proximity operators will be with us for a long time to come, however cleverly disguised.

3. 1. The classical or pre-AI situation.

In a survey of interactive retrieval systems, which was published as early as 1974 [Martin, 1974] almost all features of modern systems were already present, including searching for spelling variations and related terms capability (see figure III. 3). Indeed one might say that the only additions until well in the eighties have been the introduction of automatic relevance feedback and fuzzy logic, although [Lancaster,1972] already speculates on the possibilities of automatic relevance feedback.

Martin classifies the features of interactive information retrieval systems in groups, of which we show four that are most relevant for our discussion: SYSTEMS, INSTRUCTIONAL, QUERY FORMULATION and RESULT MANIPULATION.

In *Systems* he describes the organization of the database and the manner it is used, together with the technical tools (note the typewriter terminal, by now obsolete). Features like *full text searching* and *index searching* are of obvious importance for the possibilities listed under the header *Query formulation*, because these organizational aspects often are decisive for the form that queries may take. Readers, who are not familiar with the use of large information retrieval systems, should note the *intermediarysearcher*, a person employed by the database service, who has as task to elicit as exact as possible the information need of the user and then to translate his information need into a query that conforms to the general possibilities and the syntax of the system.

The topics described under *Instructional* are now generally considered to be a part of *userinterfacing* in the wide sense of the word (if a subdivision is to be made

SYSTEMS	INSTRUCTIONAL	QUERY FORMULATION.	RESULT MANIP.
Large textual databases	Users guide	Suffix removal	Search review
management inform.	instruction	search field control	predefined formats
full text searching	system	dictionary access	on-line formatting
index s earching	class	relational operators	rapid scan
intermediiary s earcher	personal	spelling variations	highlighting
end user searching	reading	related termcapability	expanding
video terminals	on-line	word proximity ops.	sorting
typewriter terminals	data base overview	boolean operators	ranking
telephone network	sample searches	request sets	computing microfiche
	on-line documentation	phrasedecompostition	display of graphs
	search logic training	search profiles	statistical interface
	live help	sequential searching	off-line printing
	vest pocket card		photocomposition
	comments		batch retrieval
	monitor log		data access protection

III.3. Features of IR-systems before 1974.

here, it should be made according to the sections *training* and *documentation*, the latter including the various help-strategies).

Most of the items under *Query formulation* and *Result manipulation* are very much the subjects of research in IR. Many of the topics, that are placed by Martin under *Resultmanipulation*, might as well or better be placed under *Query formulation* as these results generally act as new input to the query end. Search profiles would now be ordered under user interfaces, as they form part of user modeling.

To deepen our understanding of the features in these two groups, we may make a loose classification of retrieval tools in four new groups, adding short descriptions of the less known of them:

3. 1. 1. Word-oriented tools

These are tools that allow easy expanding or restricting of queries by manipulating the searchwords in the query. An important technique is suffix removal or truncation at *query time* (as opposed to techniques that truncate words at *indexing time*, (we will maintain the difference between *truncation*, *stemming* and *lemmatization*, the first just cutting a number of characters from the word, the second trying to remove pre- and suffixes and the third attempting to reconstruct the original lemma). Germanic languages like english, dutch or german generally profit more from these techniques than e.g. french. Also spelling variations and 'soundex' techniques, that try to find words "sounding like..." fit in this category. Sequential scanning, that allows for the retrieval of more words in a row or even admits regular expressions, is for its succes as much dependent from the size of the database as from the power of the computer.

3. 1. 2. Selectors and combination tools.

Operators that enable the user to expand or restrict the resulting sets by e.g. *boolean* operators, *proximity* operators and *relational* operators. Boolean operators in themselves will need no further explanation and neither do the relational operators (Greater Than, Smaller Than, Within Limits etc.). Proximity operators govern the distance between the occurrences of keywords in the text, both absolute (in the number of words between two keywords) and relative (the occurrence in the same sentence, paragraph or page). A noteworthy development is the attempt to add weights to the various operators in the same way as weights are given to individual keywords. We will discuss weighting in chapter VI.

Another tool is *field control*, that enables queries or the occurence of keywords to be restricted to one or more selected fields. As IR systems and FTIR systems are not as structured as normal (relational) database systems are, this tool is not so powerful as one might think.

The tools mentioned above, if used in a query, return sets of records. These sets of records may be used in several ways. Many systems keep a record of the individual queries and sets of documents resulting from them. Subsequently these systems may allow combinations of the individual sets or of sets with new keywords as new queries.

Relevance feedback may also be considered as belonging either in this group or in the next. This technique considers the (other) keywords that are attached to

INSTRUMENTS
USE EQUIPMENT

INSUFFICIENCY
USE ADEQUACY

INSURANCE
SN *
RT 13001

INTEGRATION
USE RACIAL INTEGRATION

INTELLIGENCE
SN *
UF GENERAL ABILITY
MENTAL ABILITY
RT 6001
INTELLIGENCE TEST@
MENTAL DEVELOPMENT@

INTELLIGENCE QUOTIENT
UF IQ
RT 6001

INTELLIGIBILITY
UF COHERENCE
UNDERSTANDABILITY
RT 11011
LOGICAL ORGANIZATION@
PERCEPTIBILITY@

INTERPERSONAL CONFERENCE
SN FACE TO FACE COMMUNICATION, AS BETWEEN TEACHER AND STUDENT
UF CONFERENCE (INTERPERSONAL)
RT 20187
COUNSELING@
INTERPERSONAL RELATIONS@

INTERPERSONAL RELATIONS
SN *
RT 12003
FRIENDLINESS@
NTERPERSONAL CONFERENCE@
SOCIAL ATTITUDE@
SOCIAL DIFFERENCES@
SOCIAL INTERACTION@
SOCIAL PROBLEMS@

INTERPLAY
USE INTERACTION

INTERPRETATION
SN *
RT 4001
DISRUPTIVE BEHAVIOUR

INTERSENSORY PERCEPTION
SN PERCEPTION INVOLVING SEVERAL SENSE MODES
RT 2016
MULTISENSORY PRESENTATION@
SENSE MODE

INTERSTUDENT RELATIONS
RT 12003
SOCIOMETRIC TESTR@

(from: Information Retrieval Thesaurus of Education Terms)

III. 4. Page of a thesaurus.

documents, retrieved by a set of keywords. The additional keywords then are used to reformulate the original query.

3. 1. 3. memory nudgers

A third class of retrieval tools I would like to call *memory nudgers*. They operate on the well-proven fact that recognition in humans works better than recall. This has called into being a family of tools that allow dictionary access, offer related

term capability (thesaurus, see fig III. 4) etc. Techniques like relevance feedback or truncation mentioned above, might also be classified under this header, if they manifest themselves as a part of the users interface.

It might be argued that a thesaurus, if it is not directly derived from the underlying database or at least composed for it, might as well be classified under the general header of user interfacing. This, in itself, is true enough. But a thesaurus also may be an important receptacle for the creation, storage and retrieval of knowledge about the texts, which makes it a very important retrieval tool indeed.

There are several types of thesauri. An important difference is that between *normative* thesauri and *descriptive* thesauri. The normative thesaurus describes an 'ideal' or regulative network of meanings and attain their status by fiat or agreement. They may be compared with classification systems such as are used in assigned indexing (see chapter I). The descriptive thesaurus derives its term relationships from the document representations in the IR system and is generally based on co-occurrence statistics, e.g.:

$$P(I_j/I_k) = \frac{\#\ of\ 'times I_j\ co\text{-}occurs\ with\ I_k}{\#\ of\ 'times I_k\ occurs}$$

where of course $P(I_j/I_k) <> P(I_k/I_j)$. This formula may be replaced with others, that take certain tresholds or the results from control- or calibrationdocuments in account.

A descriptive thesaurus, therefore, expresses the semantic relationships between terms as primitive, uninterpreted and symmetric (i.e. synonyms and Related term capability), while normative thesauri, such as the one from which the figure III. 4) was taken, offers the possibility of Broader and Narrower term relationships.

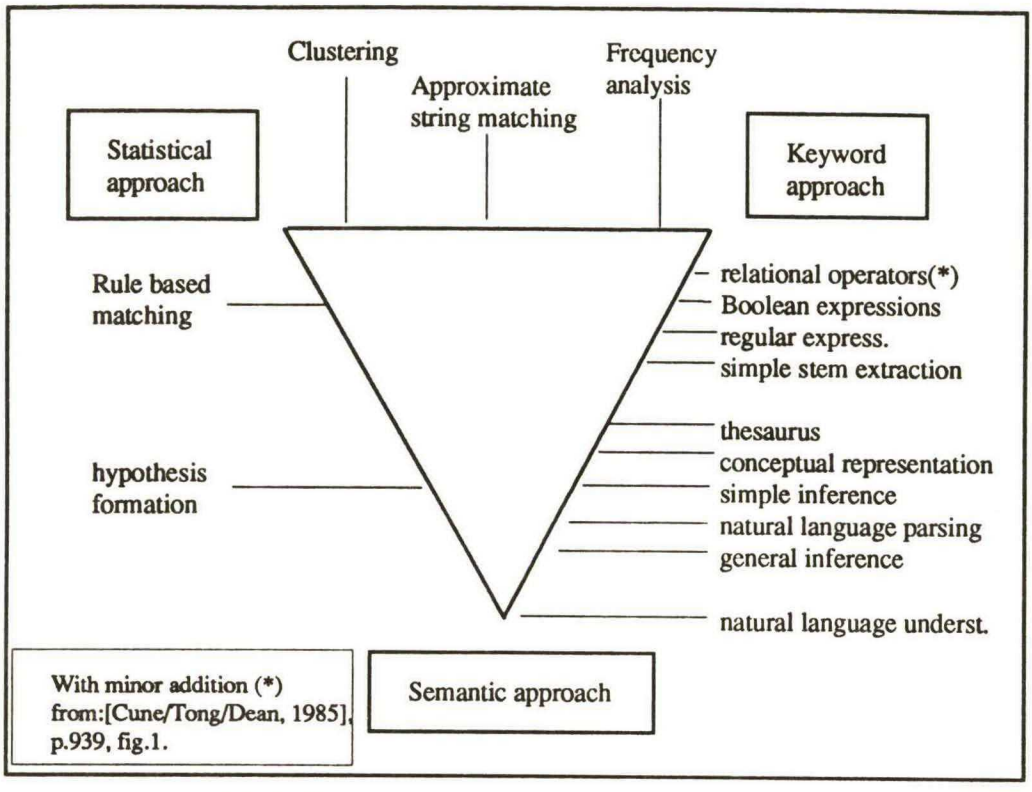
We will return to the subject of thesauri several times in this memo, notably in chapter VI.

3. 1. 4. User interfacing.

Other features, that might be classified onder the general header of user interfacing, are those tools, that collect attributes and characteristics of the individual user and use these characteristics to smoothen the use of the system (user interfacing by user modelling). See also the discussion of RUBRIC/TOPIC. More in general user interfacing concerns itself with the design of menus, command language or a natural language interface, the accessibility of on-line help, on-line thesaurus and index (as a memory nudgers) et cetera.

3. 2. The present situation and the shape of things to come.

If we compare the list of Martin with a more modern view of IR techniques as laid down in the schema by [Cune/Tong/Dean, 1985] (fig. III. 5.) we note some differences. To start with, the schema is conceptual, rather than enumerative, trying to relate three approaches in stead of summing up features of systems. The drawing connects three general approaches to IR, called respectively the *Statistical*, the *Semantic* and the *Keyword* approach, in a triangle. The concepts, that are put by Martin in the column *QueryFormulation.* are in the Cune/Tong/Dean drawing arranged along one side close to the 'Keyword corner'. The word *thesaurus* marks



III.5. The information Retrieval Triangle

the point where simple words are augmented with the relations that exist between them. From this point on the influence of semantics becomes stronger up to the hypothetical point that information retrieval works with full natural language understanding of the documents.

Again starting from the Keyword approach, but working to the left, ever increasing statistical processing may be applied to the original document. Frequency analysis and similar techniques as described in this chapter may be used to generate inverted files of keywords (that subsequently may be retrieved by the techniques below the keyword corner). Statistical processing may lead to the comparision of documents on the grounds of the keywords, thus creating clusters of documents that score above a treshold of similarity (also described below). From the left corner down to the Semantic approach, two techniques are mentioned that fall outside the literature studied so far.

The observer will notice that the upper and the left side of the triangle are very sparsely populated, compared to the long list of features that covers the right side of the triangle. Even if considered that most of the features below *thesaurus* are conjectural or do exist only in experimental systems, it is clear that Information Retrieval still is very much keyword-oriented, although in the weighing of the keywords generally statistical information is used in a greater or lesser degree.

4. Measuring retrieval performance.

It is important to establish procedures and rules to compare the performance of information retrieval systems. For highly formalized systems like data retrieval systems (e.g. relational database systems) this poses no particular problems, or better said: the problems in measuring those systems are fundamentally different from those encountered in deciding on the performance of IR-systems. As we have seen, information retrieval systems typically have to cope with highly imprecise concepts like the 'aboutness' of a document and 'user satisfaction'.

Let us have a look at a typical IR-system using keywords. It generally will consist of a database with information about documents (i.e. bibliographic lists, abstracts or even full-text documents). Apart from that there will be one or more inverted files with keywords, acting as secondary keys. The individual entries in this files are descriptions of concepts that are found in the documents and which serve as signposts to the documents. In manual systems every attribute will have a separate file, in automatized systems they usually are merged into one.

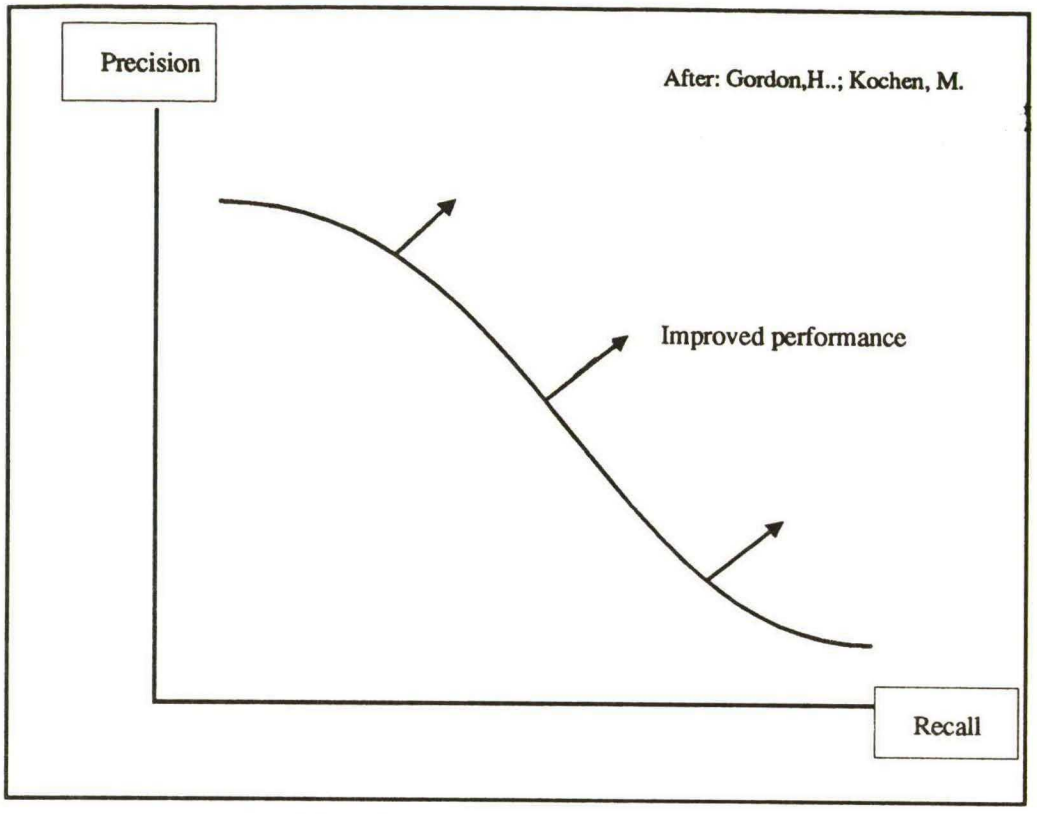
4. 1. The Prediction Criterion and the Futility Point.

Now if a user approaches the system, he generally will try to predict which word or term is used to describe the documents that he is interested in. Then he will check the appropriate entrances and if that term is an entry, he will (possibly after several rounds of specifying and combining results) inspect the documents, or rather the description of the document that exists in the database (the on-line document). If he cannot find the entry, he will choose other terms to describe his information need or decide that no documents in the database will satisfy his needs and consult another library.

Therefore the first property of an IR-system should be the predictability of the keywords, terms or structures, which represent the documents in the system. This is called the *Predictioncriterion* and it is one of the requirements for succesful retrieval. However, even if a user succeeds in predicting in which way a sought document is represented in the system, it is only a part of the way towards succesful retrieval.

For example: consider the document D_i, indexed with the keywords K_a,K_b,K_c,K_d and K_e. Before an user may retrieve that particular document he has to predict the fact that at least one of those keywords is used to represent the original document. Now let us assume that he chooses K_c to use in his query and further that the keyword K_c is assigned to a hundred documents in the database, but that the user is only interested in D_i. The result of query K_c will therefore result in a hundred retrieved documents, which he has to look through in order to select the one document D_i he really wants. Possible he is not willing to spend the time needed to browse through the documents and stops halfway before finding D_i or even does not start at all. Either way the net result of the query is zero.

This shows us that predicting the correct keyword is not sufficient for succesfull retrieval, but that another factor or factors come into play: the *futilitypoint* (FP): the number of documents after which the user stops browsing through the



III.6. Precision-recall trade-off.

documents retrieved. A refinement is the *anticipated futility point*; the number of documents, so big that the user not even begins browsing through with the net result that the document will not be retrieved *even if it would have been the first document to be displayed!*[Blair, 1980].

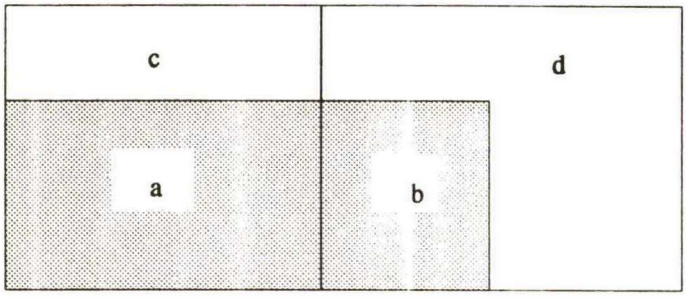
Much more may be said about the PC and the FP. Suffice it to observe that the psychological processes of the user are as important for the succes of an IR-system as the technical performance of the system. Therefore the ease and clarity with which these document representations and the on-line documents may be consulted and inspected are exceptionally important.

In Gauch [Gauch&Gauch, 1990] the number of separate questions to be asked has been choosen as a measuring criterion for system performance.

4. 2. Precision and Recall.

Traditionally the performance of an IR-system is expressed in two measures: the precision and the recall. After a query the *precision* is the ratio between the relevant documents in a batch of retrieved documents and the total number of documents that were retrieved. The *recall* is the number of relevant records retrieved as compared to the (estimated) number of relevant records in the database (see fig. III. 7.).

	Users appreciation relevant	not relevant	Total
Found	a (hits)	b (noise)	a + b
Not found	c (misses)	d (rightly so)	c + c
Totaal	a + c	b + d	a + b + c + d
			(total database)



III.7. Hits and misses after retrieval.

$$precision = \frac{number\ relevant\ \text{and}\ retrieved}{total\ number\ retrieved}$$

$$recall = \frac{number\ relevant\ \text{and}\ retrieved}{total\ number\ relevant\ in\ database}$$

The typical relation between precision and recall is shown in figure III.5. Movement of this line upward and to the right is an indication of improved system performance.

Of course there is more to system evaluation than just precision and recall. Salton [Salton/McGill, 1983] mentions six critical evaluation criteria:

1. The *recall*, that is, the ability of the system to present all relevant items.
2. The *precision*, that is, the ability to present only the relevant items.
3. The *effort*, intellectual or physical, required from the users in formulating the queries, conducting the search and screening the output.
4. The *time* interval which elapses between receipt of a user query and the presentation of the system responses.
5. The form of *presentation* of the search output which influences the user's ability to utilize the represented materials
6. The collection *coverage*, that is, the extent to which all relevant items are included in the system.

Salton continues by remarking that "*...of the six user criteria all but two are relatively easy to measure.(...) This leaves us the recall and precision measures.*" However, many recent studies seem to contradict this. The human factors involved in criteria like *effort* and *presentation* are notoriously difficult to evaluate, nevertheless they have an important role in overall system performance and may directly influence criteria like recall and precision.

It is clear that human factors like the effort needed in formulating refinements of the queries or presentation of the output may strongly influence the performance of the system as a function of user satisfaction.

5. Early index-based models.

We have already observed the fact that the BOB-index, or at least an index that consists of a list of keywords (key-phrases) and place-indicators (pages, documents) is typical for most IR-systems. Blair [Blair, 1990] distinguishes twelve possible models for such systems (fig. III. 8), but as all twelve already had been implemented before 1980, we will call them the *early models*.

In the figure we have presented the models in the same order as Blair did. Although we feel that Blair's representation is incomplete and a bit lop-sided, we will discuss it in some length and add some observations.

5. 1. The twelve models of Blair.

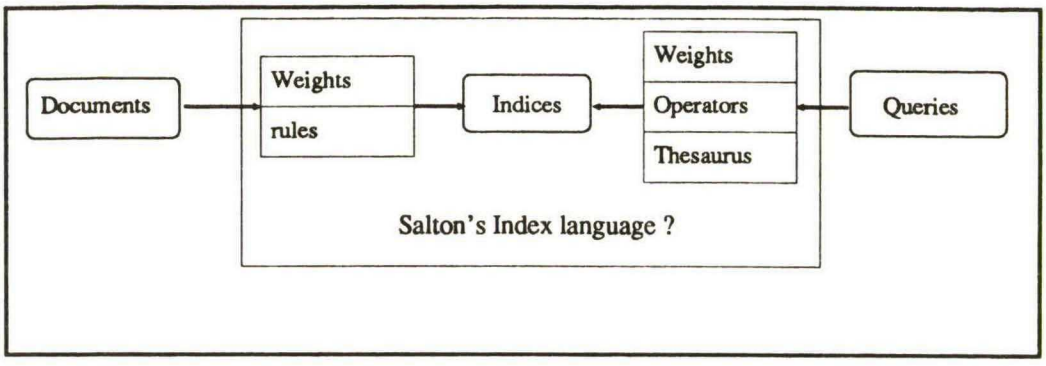
If we compare Blair's models with the general model of Salton (chapter II) or with the scheme of Tong etc. (fig. III. 5.) we feel that Blair attempts to come to grips with the possible variations in the relations between the query and the index language. In his models the documents are presented by document surrogates which in their turn consist of sets of descriptors (or possibly the complete document). The way that the document surrogate is arrived at, whether by assignment or by derivation; if assigned, according to which system and if derived, by what algorithm, is not taken in account, although this may well be the single most important problem of IR.

If we take a closer look at these models, we can see that model II and model IX are equivalent. As Blair himself observes, model II is a complete boolean system, as disjunctive queries can always be transformed into conjunctive queries[4]. It is not clear why he has allowed model IX to remain in his list, except to suggest a better interface, which can solve similar boolean puzzles and which can offer other enhancements. In his discussion of model IX he mentions such enhancements: feedback techniques, associative searching procedures and term weighting schemes. Again it is not easy to see why concepts like term weighting in models V, VI and VII are important enough to claim separate models when they occurr in the query, the Index record or in both, while in model IX they just are mentioned as possibilities.

4 E.g.: p . (q v r) may by DeMorgan's theorem be rewritten as (p . -(-q . -r), thus eliminating the need for the v-operator.

Model	Request	Index record	Retrieval	Retrieval rules
I	Single descriptors	Set of one or more descriptors	Weak ordering (yes or no)	If the descriptor in the request is a member of the descriptors assigned to the document.
II	Set of descriptors	Set of descriptors	weak	If ALL the descriptors in the request are in the index record
III	Set of descriptors + 'cut-off' value	Set of descriptors	weak	if it shares a number of descriptors with the request that exceeds the cut-off value (quorum).
IV	Set of descriptors + cut-off value	Set of descriptors	ranked	As model III, but ranked by overlap.
V	Set of descriptors each of which has a positive number associated.	Set of descriptors	ranked	Documents are ranked by the sum of the weights of descriptors common to the request and the index record.
VI	Set of descriptors	Set of descriptors, each of which has a positive number associated	ranked	Documents are ranked by the sum of the weights of descriptors common to the request and the index record.
VII	Set of descriptors each of which has a positive number associated.	Set of descriptors each of which has a positive number associated.	ranked	Documents are ranked by the sum of the products each of which results from the multiplication of the weights of descriptor in the request by the weight of the same descriptor in the index record.
VIII	Set of descriptors each of which has a positive number associated.	Set of descriptors each of which has a positive number associated.	ranked	The weights of the descriptors common to the request and an indexing record are treated as vectors. The documents are ranked according to the cosine of the angle between the vectors.
IX	Any boolean combination of the following forms: Intersection, Union or Negation.	Set of descriptors.	weak	Records of which keywords occurr in the resulting set are retrieved.
X	Any boolean combination and/or proximity expression.	Entire text is searchable	weak	Records of which keywords occurr in the resulting set are retrieved
XI	Single descriptors	Set of descriptors	weak	Request term is expanded by a thesaurus.
XII	Single descriptors	Set of descriptors	weak	Request term is expanded by a thesaurus, taking degrees of semantic relatedness .

III.8. Blair's models.



III.9. Alternative model for early IR.

An enhancement that *does* claim two more models is the thesaurus, used for either straightforward expanding of the original query (model XI) or for expanding after reaching a treshold (model XII).

The exact reasons for Blair's scheme thus is difficult to comprehend, so we will try to re-arrange his models in a more general scheme (fig. III. 9), which emphasizes the similarity with Salton's more general model. To start with we see how rules are used to create indices from the original documents. Blair does not give an indication of the many different ways in which documents may be processed to fill indices (except the special case of the full text document), but he does mention the possibility of adding weights to those indices (models VI-VIII). Suffice to say that many strategies for extracting keywords or assigning them from classification systems were developed in the sixties and well-known in the seventies and eighties (see [Paice, 1990], discussed in chapter VI). On the other side of the figure we see how the query is matched with the indices: using operators to combine several keywords, attaching weights to keywords to indicate relative importance and by using thesauri to expand sets of keywords. The aspect of the user interface is conspicuous by its absence, which is indeed amazing if one takes in consideration that Blair is one of the protagonists of the futility point and the predicted futility point and certainly is aware of the great influence that the user interface has on such variables.

IV. The documents.

1. Document types.

Our main interest is centered on natural language texts (documents) as containers of info and on the selection and retrieval of those texts, which will satisfy the information need of the user. In the last chapter, about databases, we have surveyed the orthodox access methods to texts. Apart from the normal keys as author, title or publication, these methods use keywords and keyphrases as secondary keys, either assigned according to a classification system or selected from the text on one criterion or another.In this chapter we will consider what exactly constitues a document in its environment and what may happen to a document when it comes in the orbit of a IR system.

1. 1. What is a document.

What may be considered a document in an IR system? Most authors take a document for granted and do not really try to define it. Faloutsos starts defining a document as "*a piece of text without attributes*" ([Faloutsos, 1986], p.49), or rather, he explicitly ignores the existence of attributes except for the last section of the paper (the word 'attribute' here obviously has to be taken in its meaning of a slot for a value in a database and not as a visual property of a text).

"*IR is concerned with the representation, storage, organization and accessing of information items. In principle no restriction is placed on the type of item (...) In actuality many items (...) are characterized by an emphasis on narrative information.*" ([Salton&McGill, 1983] p.1-2). I think it is safe to read document for 'narrative information'. He also professes the bias towards verbal information of written documents as opposed to e.g. visual information of pictures.

We will take the stand that a document is primarily a cohesive body of written natural language without explicit structure, although a structure of one kind or another will generally be inherent or imposed on it, e.g. the chapter-paragraph-sentence structure (we will call this the the table of contents or TOC, although the TOC actually is a subset of chapter- and paragraph headings). Of course, documents display other properties, which we will cover more extensively in the next chapters. Also document-structures are described, that try to define connectivity of adjacent phrases and sentences, or pragmatical patterns (for abstracts: [Liddy, 1988] or even text grammars [vanDijk,...]). Here we will be concerned with groups of documents and with the nomenclature surrounding them when stored in an IR-system.

The minimal condition for a document seems to be, that it consists of one or more strings with words, the primary purpose of which *has been* to convey a meaning

Ond.werp	gezegde	lijd.vw	vz1	object1	vz2	object2
mensen	beschrijven	gebeurtenissen	in	zinnen		
mensen	registreren	zinnen	in	tabellen	met	sql
mensen	definieeren	tabellen	in	computer		
mensen	ontwikkelen	systemen	voor	tabellen	in	computer
mensen	vragen	informatie	uit	tabellen	met	sql

IV.1. Tuples in GRAMMARS.

or purpose to a human reader; therefore we will use the *locutionary*, *illocutionary* and *perlocutionary* acts from Austin [Levinson, 1983] and apply them to documents[1]. The italics of ***has been*** implicate that this meaning may be lost in the document itself, as we will see when discussing corpora. Neither does this string have to be a syntactical well-formed expression.

1. 1. 1. Sublanguages

This last relaxation of the definition of natural language is necessary for two reasons. The first is that many documents, e.g. shorthand descriptions of objects in a museum database or notes jotted down in interviews, e.g. [Liddy, 1991], may incorporate utterances in a sublanguage, which conflict with normal usage (i.e. are not *well-formed* according to a NL grammar). The same is true for titles of books, articles or pictures.

The second reason may be found in the fact that some documents may be written in a formal language. Computerprograms fall in this category. Also a group of formal 'languages' is emerging, which mimic simple sentences in natural languages. In some of these systems relational databases are constructed in which the attributes have the function of phrases resembling NL phrases and the domains contain expressions, which have the same form as NL expressions. The intension of the database is the grammar for such languages. An example is GRAMMARS [Dijk&Swede&Visser,1989],fig.IV.1. The goal of these systems is the formalizing of certain relations in a domain in such a way that users, which are not aware of the formalization and indeed would not know how to apply it consciously, are nevertheless capable of working with it. Of course this may be considered as just another example of a sublanguage. We will call these languages pseudo-NL languages.

The fact that this kind of documents does not contain natural language, does not alter the fact that the primary purpose of both computerprograms and pseudo-NL languages is the conveyance of meaning to the human reader and that information retrieval may be applied to them. This in contrast with the orthodox relational

1 locutionary act: the utterance of a sentence with determinate sense and reference.
illocutionary act: the making of a statement by virtue of the conventional force associated with it.
perlocutionary act: the bringing about of effects on the audience by means of uttering the sentence.

03 gezien 600 de 370 lange 103 duur 000 van 600 vele 453 verzekeringscontracten 001 is 273 dit 360 onvermijdelijk 100 , vooral 500 omdat 710 de 370 aard 000 van 600 deze 370 contracten 001 een 450 tus sentijdse 103 premieverhoging 000 niet 500 toelaat 253 .
04 de 370 premieverlaging 000 geldt 243 , zoals 710 onlangs 500 reeds 500 werd 275 aangekondigd 216 , voor 600 nieuwe 103 verzekeringen 001 , gesloten 216 vanaf 600 15 470 september 010 jl. 100
05 het 370 loslaten 211 van 600 de 370 vaste 103 wisselkoers 000 van 600 de 370 duitse 103 mark 000 heeft 273 geleid 206 tot 600 een 450 ernstige 103 botsing 000 tussen 600 de 370 europese commissie 01 0 en 700 de 370 regering 000 van 600 de 370 bondsrepubliek 000 .

IV.2. Part of the Eindhoven Corpus.

databases, where the contents first serve the efficiency of combining and relating tables; intelligibility for the human user coming second. Nevertheless this does not diminuish the perlocutionary force of statements in these pseudo-languages.

1. 1. 2. Corpora.

The *'primary purpose to convey a meaning'* as mentioned above applies to the utterance in its original context. If it is taken out of this context, this meaning or purpose may be lost: the semantics of the document have no perlocutionary force. However, a collection of unconnected sentences, e.g. a corpus for linguistic research (fig. IV,2) may still be considered a document. In this case the sentences are themselves well-formed, but are not connected to convey an overall meaning. Ideally an IR-system should be able to flag such a corpus as having an unusual, contradictory or impossible content, but still be able to extract information from it. At first glance information retrieval in such a context does not make much sense. But the same properties that make such a corpus appropriate for research in linguistic phenomena, may very well serve to use such a corpus in other research, in which IR may be an important asset, either as a tool or as the subject. Given that a corpus is a cross section of utterances of a certain group of language users, it may as well be used to gauge the frequency with which that group mentions e.g. food or the second world war, as the frequency of particle-noun combinations or the character 'e'. In the first case an IR-approach and -techniques are perfectly valid.

1. 1. 3. Normal communicative text.

The bulk of the texts in an IR system consists of normal, wellformed expressions in a natural language or a habitable subset of a NL. Nevertheless there are a number of differences between texts: e.g. poetry as opposed to prose, narrative texts vs. informative or documentary text. IR by its nature is aimed very much at informative text; however, there is no reason why narrative texts or poems could not be entered in an IR system. In this electronic age we should also be alert to

the possibility that the document, which we see on the terminal or of which we may have a paper copy, has been pasted together from boilerplate and data from a database: therefore has never existed before and may never exist again. The paper copy therefore may be the only instance of that particular 'document', even though the system that spawned it still may be active. Of course a document, that only existed on the screen of a terminal is even more ephemeral.

A text may be seen as an aggregate of many structures: as sequence of tokens (characters, keywords) and sentences, up to the macrostructure of chapters and possibly volumes and all the other structures that may be found in the typical table of contents. We will call this kind of structures *document representations.*
A text is also a collection of meaningful clusters of statements and facts, to be combined to information or knowledge. Browsing through a corpus as in the figure above, one might find many such facts and statements, isolated from its information/knowledge context. Nevertheless they can easily be recognized as such. Both these smaller statements or propositions and the greater structures that can be recognized using pragmatic knowledge or text grammars, if they are made explicit and stored, will be called *document knowledge representations.*

Although in texts several structures may be recognized by the human user, the value of these structures for Information retrieval may vary considerably. Also in many cases it is next to impossible to extract these structures by computer, which is paramount to saying that there is no consistent way of recognizing and isolating them. On the other hand those structures that are readily recognized by computers often do not add substantially to succesfull extraction of information
For instance: a computer is adept in counting the characters in a document and ordering the results in a frequency list. The value of those lists for purposes of IR, while not exactly zero, is not very high. The counting and ordering of word tokens and -types fares significantly better in that respect; however, this approach also breaks down when the number and length of the documents increase. The juggling with relative document frequencies, comparative corpora and other probabilistic approaches (see [Salton&McGill, 1983] and [Rijsbergen, 1979]) may push the point of ultimate failure farther up, but it does not substantially alter the picture.

Much research has been done on Document Knowledge-like representations, but no substantial progress has been made towards automatic extraction of those representations. Also, they often seem to concentrate on the progress of dialogues, rather than on the topicality or "aboutness". This especially true for theories as the Rhetorical Structure Theory [Mann&Thompson, 1987] or the theories of Barbara Grosz [Grosz&Sidner, 1986]. Older, but still valid are the observations of Schank [Schank&Abelson, 1977] on frames and acts, giving birth to attempts to convert sentences into primitive acts, thus effectively paraphrasing these sentences. This usually involves the creation of larger and lower-level descriptions than the original sentences. Working in the opposite direction we mention Lehnert's Abstraction units, that convert descriptive nets into smaller, less detailed and

higher level ones (see [Winston, 1984]). We will go into these theories more extensively later.

The group of internal or data properties, if not conductive to 'understanding' documents, is at least easier to describe and code. We already did mention 'countables' such as characters and words, the organizing and retrieval of which are by far the most used tools of information retrieval. Less important, but also easily detected and coded are the properties of chapter-paragraph and section structure (TOC), lay-out, attribution to an author etc., in short all those properties, which may be described in a mark-up language such as LATEX or SGML. An exhaustive survey of these properties and proposals for encoding them is to be found in the work of the Text Encoding Initiative [Burnard, 1990] (see chapter V).

To wrap it all up: documents are (written) collections of utterances that native speakers understand, either on the semantic level (corpora), the pragmatic level (sublanguages) or both (normal communicative texts). Documents may be stored individually or as parts of bigger files or clusters of files. The organisation of these files may or may not be meaningful in relation to the information retrieval activity: in any case the information related to the storage may be considered as a part of the internal information of the document. Also the IR-system will generally build a number of auxiliary files as a result of processing the documents.

2. Documents in the system: some definitions.

In this section we will consider a nomenclature of the documents, or parts of documents, that will be used in IR-systems.

2. 1. Document surrogates

Any processing of documents in such a system will have to start with the decision how the original document will be presented to the system, i.e. the total text of the document, or an abstract or just a bibliographic record with some keywords. This 'document' that is entered in the system we will call the *document surrogate*. Obviously the document surrogates that are presented to the system should all be of the same kind.

2. 2. Document representations

The system will create from this document surrogate one or more representations of the document, which we will call the *document representations*, abbreviated to *docreps*. These may be either *Data representations* or *Document Knowledge representations* as discussed in chapter II. If the document surrogate only existed of the bibliographic reference and keywords, this processing will be limited to the insertion in a database and updating of the lists with keywords. Document surrogates in the form of the abstract or the total document may need considerably more processing before the document representations are created.

These representations will be used to represent the document in the system when the similarity functions are applied to the docreps and the relevant representation of the query.

2. 3. Additional information.

Additional information (also called formal or bookkeeping type information) may be added to the representation(s) of the document in the database to account for essential information, which is not in the text proper or to facilitate processing. This information is often stored separately, e.g. in the fields of an orthodox database and it may represent the document at retrieval time.

2. 4. The online document

The information accessible when the document is 'found' we will call the *on-line document*. In some systems this might include the WYSIWYG[2] representation of the original document, but ultimately it may be any part or combination of the representations and the document surrogate.

An abstract or an hierarchically structured table of content (TOC), as well as other identifiable components may already be part of the document surrogate; when they are generated as one of the results of the processing in the system and subsequently are stored in the system, they rightly should be called document representations. Non-linguistic documents (images) are not considered here, except in sofar as they exist in the text of a natural language document, e.g. captions, references and/or verbal descriptions.

A special case is that when the on-line document, the document surrogate and the document representation are identical. This often is the case in systems that use full text scanning to retrieve documents, e.g. in searches using *grep*, the well-known pattern-matcher from the UNIX-environment.

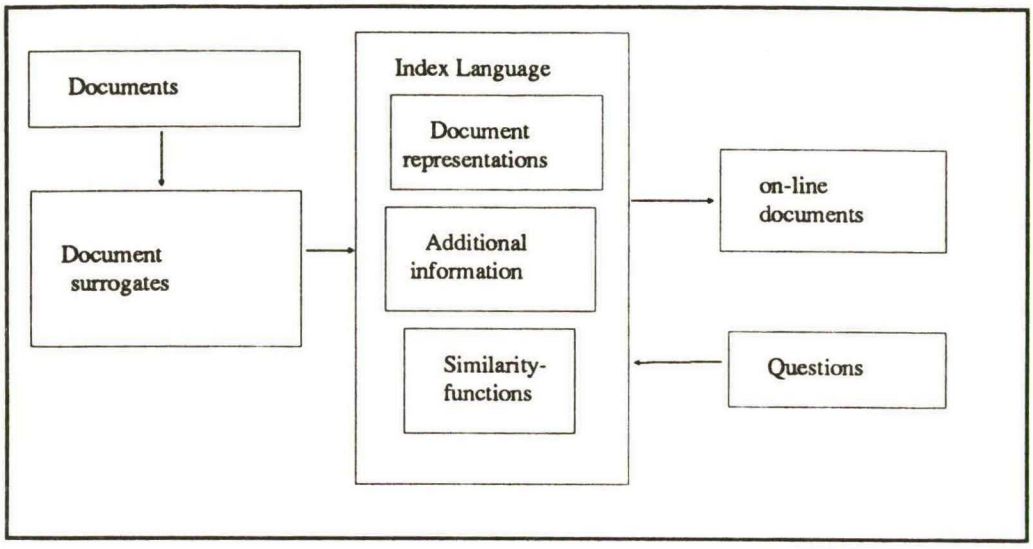
2. 5. Abstracts and extracts.

In publications on Information Retrieval the division between this *document surrogate*, the *documentrepresentation* and the *on-line document* is not always clear. Especially there exists some confusion about the exact place of abstracts in the scheme of things.

Let us first decide that abstracts and extracts are two totally different animals. Although both serve to represent the original document,"*...so that the reader may decide, quickly and accurately, whether they* (sic) *need to read the entire document.*"([ANSI. Z39-14], p.1), they are generated in totally different ways. Thus an extract is a part or parts of the original, selected to represent the whole and it consists of selected sentences from the original. An abstract, on the other hand, is an independent description of an internalization of the original document. The ANSI text defines an abstract as "*...an abbreviated, accurate representation of the contents of a document.*" We do prefer the word *internalization* in stead of *contents* because it puts emphasis on the fact that a processing and reformulation of the document is a prerequisite for the generation of an abstract. Examples of this internalization may be found in [Lebowitz, 1986].

---

2 WYSIWYG: What You See Is What You Get. Originally used to indicate wordprocessors that produce facsimiles of the ultimate paper output on the screen.



IV. 3. Extended Information Retrieval model.

Paice [Paice, 1990] gives a short but concise description of the attempts to create abstracts or rather *extracts* automatically: the generation of these extracts then is very much a part of the indexing process and in that case we will not consider them document surrogates. Alternatively the indexing process may use an existing abstract (manually generated or included by the author) in stead of the document itself. In that case the abstract effectively becomes the document (document surrogate), even if the whole document is displayed at retrieval time (the on-line document). We will consider techniques to create abstracts and extracts automatically in chapter VI.

To conclude it all we will look at the progress of a document through an IR-system in a library and so demonstrate the different terms. We will take a paper document as an example, although with the evolution of electronic mail many documents never see a printer during their lives.

The first act of the librarian will be to decide what will be fed to the indexing part of the system (document surrogate) and what will constitute the on-line document as the last stage of the retrieval process. He might do an optical scan of the document and store a facsimile electronically or on micro film. This facsimile might be processed by an optical character reader (OCR) to obtain a machine-readable representation of the text, with or without mark-ups.

It depends of the sophistication of the system whether human assistance is needed after a machine-readable document surrogate is obtained. Ideally of course, the system should be able to continue on its own, extract keywords and other information and organize this information for retrieval. The surrogate effectively becomes the document itself; the internal representation is generated by the system. It is as yet more common that a human indexer will first read the document and

fill in those parts of the document representation which the system is not able to generate itself. Typically the bibliographic data are extracted along with a number of keywords, which to the indexer best represent the contents of the document. This information then becomes the document surrogate, which the system stores without extended additional processing.

At retrieval time the user tries to select among the document representations those of which he expects that the documents concerned will satisfy his imformation need. After he has made a selection, but before he leaves the system, he may want to check the retrieved references. The information, which at that point is available to him we call the on-line document.

1. Bibliography

ANSI:" American national standard for writing abstracts." ANSI Z39.14- 1979

Attig, J.C.:" The concept of a MARC format" Information technology and libraries 2 (March 1983): p.7-17

Baars, C.G.H.; Schotel, H. :" Natuurlijke taal en databases" A.I.T. 1988

Bar-Hillel, Y.:" The mechanization of literature searching." in: National Physical Laboratory: Proceedings of a symposium on the mechanization of thought processes 2, 1959

Bates, M.A.:" Information search tactics." Journal of the American society for Information Science 1979, No 4. 204-214

Baxendale,P.b. :" Man-made index for technical literature - an experiment. " I.B.M. jopurnal of research and development,2 pp.354-361; 1958

Benschop, C.A.; Heer, T. de:" Voortzetting van het informatiesporen onderzoek" IWIS/TNO 1980

Blair, D.C. :" Searching biases in large interactive document retrieval systems." Journal of American Soc. for Information science, 31:4. p.271-277, 1980

Blair, D.C.:" Language and representation in information retrieval" Elsevier, Amsterdam 1990

Blair, D.C.; Maron, M.E. :" An evaluation of retrieval effectiveness for a full-text document retrieval system" Communic. of the ACM V28:3 pp.289-299, 1985

Borko, H.; Bernick, M.:" Automatic document classification." Journal of the association for computing machinery, 1963, p.151-162.

Brauen, T.:" Document vector modification" in: Salton(1971) :509:

Burkowski, F.J. :" The use of retrieval filters to localize information in a Hierarchically tagged text-dominated database" RIAO91, p.264 - 284

Burnard, L. :" The text encoding initiative " Oxford University computing service g.j.

Codd, E.F.:" Normalized data base structure; a brief tutorial" Proceedings of ACM SIGFIDET workshop on data description access and control 1971 pp. 1-16

Chudacek, J.:" Least effort text-retrieval definitiestudierapport." IWIS/TNO 1983

Chudacek, J.:" Statistische en organisatorische eigenschappen van trigrammen in natuurlijke talen." IWIS-TNO Den Haag 1983

Cleverdon, A.W. :" Optimizing convenient on-line access to bibliographic databases" Inf. Serv. Use 4 pp.37-47, (1984)

Cleverdon, C.W.; Keen, E.M. :" Aslib-Cranfield research project" Cranfield institute of technology, Cranfield, England 1966

Conklin, J.:" Hypertext: an introduction and survey." Computer, September 1987, p.17

Daelemans, W.:" Studies in Language technology: an object-oriented computermodel of morphological aspects of dutch"

Date, C.J. :" An introduction in Data-Base Systems " Addison-Wesley, 3th ed. 1981

Davies, R.:" Classification and ratiocination: a perennial quest." Davies '86 :745:

Davies, R.:" Outlines of the emerging paradigm in cataloging" Information Processing and Management 23(2),p.89-98, 1987

Davies, R.D.:" Intelligent information systems; progress and prospects." Horwood ltd. Chicester 1986

DeJong, G. :" An overview of the FRUMP system"

Dijk, A.; Swede, V. van; Visser, J.S. :" Taalkundige informatiesystemen ontwikkeld met GRAMMARS" Pandata, Rijswijk, 1989

Earl, L. :" Experiments in automatic extracting and indexing " Information storage and retrieval 6 pp.313-334. 1970

Edmundson, H. P. :" New methods in automatic abstracting" Journal of the ACM 16, pp.264-285; 1969

Enser, P.G.B.:" Experimenting with the automatic classification of books" Aslib 1985 :650: p.66-83

Evans, D.; Ginter-Webster, K.; Hart, M.; e.a.:" Automatic indexing using selective NLP and first-order thesauri" Manuscript 1990 (?)

Evans, D; e.a. :" Computational-Linguistic approaches to Retrieval and Indexing of Text: The CLARIT project. " Carnegie Mellon University 1989

Evans, David :" Natural Language in Document Retrieval Systems (Abstract) " ITK Colloquium Series March 26, 1991

Fagan, J.L.:" The effectiveness of a nonsyntactic approach to automatic phrase indexing for document retrieval" JASIS 40(2):115-132, 1989

Faloutsos, C.; s. Christodoulakis:" Signature files: an access method for documents and its analytical performance evaluation" Transact. on Office Autom. systems, Vol.2 No.4, Oct.1984, pp 267-288

Faloutsos,C. :" Access Methods for Text " ACM Computing Surveys (New York, NY) 17 (1986.03) nr.1 p.49- 74 (100 refs.)

Files, J.R.; Huskey, H.D.:" An information retrieval system based on superimposed coding" Proceedings Fall Joint Computer Conference; reston Va. 1969

Foskett, A.C.:" The subject approach to information" London 1969, 4th edition 1982

Gordon, M.; Kochen, M.:" Recall-Precision trade-off: a derivation" Journal of the American society for information science 40(3):145-151, 1989

Graesser, A.C.; Black, J.B. (ed):" The psychology of questions" Lawrence Erlbaum 1985

Grosz, B.; C.L. Sidner :" Attention, intentions and the structure of discourse " Computational linguistics vol 12. Jul-sept 1986

Guthe, C.E.:" The management of small history museums" Nashville 1964

Hahn, U.:" Topic parsing: accounting for text macro structures in fulltext analysis-" Information Processing and Management vol.26, p.135-170, 1990

Hahn, U.; Reimer, U.:" Knowledge based text analysis in office environments: The text condensation system TOPIC" Lamersdorf (ed): IFIP conference proceedings of 1987, North Holland 1988 :687:

Harrison, M.C.:" Implementation of te substring test by hashing." Commun. of the ACM 14. Ppp. 777-779, 1971

Holstege, M.; Inn, Y.; Tokuda, L. :" Visual parsing: an aid to text understanding" RIAO91, p.175-193. 1991

Jolley, J.L.:" Information Handling: Einfuehrung in die Praxis der Datenverarbeitung" Fischer Taschenbuch Verlag 1968

Kieras, D.E.:" Thematic processes in the comprehension of technical prose" Britton&Black, p.89-108 :891:

Lancaster, B.C. :" The measurement and evaluation of library services" Washington 1977

Lancaster, F. W.: "Vocabulary Control for Information Retrieval", Washington D.C. 1976

Lebowitz, M.:" An experiment in intelligent information systems: RESEARCHER" Davies :745:

Lee Pao, M.; Worthen, D.:" Retrieval effectiveness by semantic and citation searching" JASIS 40(4):226-235, 1989

Levinson, S. :" Pragmatics." Cambridge university press 1983, 1984

Liddy, E.:" Structure of information in full text abstracts." RIAO88

Liddy, E.:" Sublanguage grammar in natural language processing" RIAO91, 1991, p.707-717

Loucopoulos, P.; Layzell, P.J.:" Improving information system development and evolution using a rule-based paradigm." Software engineering journal 1989, p.259-276

Luhn, H.P.:" The automatic creation of literature abstracts." I.B.M. Journal of research and Development 2(2), 159-165. 1958

MacLeod, I. A.:" Storage and retrieval of structured documents" Information processing and management, vol 26.2,pp 197-208, 1990

MacLeod, I.A.; Reuber, A.R. :" The array model: a conceptual modeling approach to document retrieval" JASIS 38(2=3):162-170, 1987

Mann, W.; S. Thompson :" Rhetorical structure theory: a theory of text organisation.reprinted from 'the structure of discourse'. " Information Sciences institute. USC 1987

Maron, M.E.:" Automatic indexing: an experimental inquiry" in: Journal of the association for computing machinery, 1961, p.404-417

Martin, T. H.:" A feature analysis of interactive retrieval systems" Stanford university California 1974

Mc Cune, B.P.; Tong, R.; Dean, J.;e.a.:" RUBRIC, a system for rule-based information retrieval" IEEE transactions on software engineering. 1985, p.939-944

Olle, W.T.:" The Codasyl Approach to Data Base Management " John Wiley & Sons Chichester 1980.

Oswald, V.A.:" Automatic indexing and abstracting of the contents of documents." Los Angeles, Planning Research Corporation, 1959

Paice, C.D.:" Constructing literature abstracts by computer: techniques and prospects." Information processing and management 26, 1990

Paijmans, J.J.:" Free text data bases" Proceedings RIAO88, Mass. 1988.

Rau, L.F.:" Conceptual information extraction and retrieval" RIAO88 Cambridge mass. M.I.T. 1988

Rau, L.F.; Jacobs, P.S. :" Natural lanuage techniques for intelligent information retrieval" SIGIR(1988):642:

Rau,L.F.; P.S. Jacobs and U. Zernik. :" Information Extraction and Text Summarization Using Linguistics Knowledge Acquisition " Information Processing and Management (Oxford) 25 (1989) nr. 4 p.419-428 (20 refs.)

Reynolds, D.:" Library automation: issues and applications." Bowker, New York, 1985

Rijsbergen, C. J. van:" A non-classical logic for information retrieval" The computer journal 29, pp.481-485. 1986

Rijsbergen, C.J. van:" Information Retrieval" Butterworths, sec. edition 1979

Rouse, W.B.; Rouse,S.H.: "Human information seeking and design of information systems", in:Information processing and management, vol.20; p.129-138, 1984

Ruge, G.; Schwarz, C.; Warner, A.: "Effectiveness and efficiency in natural language processingfor large amounts of text." in: Journal for the american society for information science 42 (6): p.450-456, 1991

SARACEVIC, T.; P. Kantor; A.Y. Chamis :" A Study of Information Seeking and Retrieving; Part 1, 2, and 3 " Journal of the ASIS (New York, NY) 39 (1988.05) nr.3 p.161-216 (with refs.)

Sager, N.:" Natural language information processing; a computational grammar of english and its applications." Addison Wesley 1981

Salton, G. :" Another Look at Automatic Text-Retrieval Systems " Communications of the ACM (New York, NY) 29 (1986.07) nr.7 p.648-656 (21 refs.)

Salton, G. ;M.J. McGill :" Introduction to Modern Information Retrieval " New York [etc.] : McGraw-Hill, 1983. - 448 pp.

Sandore, B.:" Online searching: what measure satisfaction" Aslib 1990 (?)

Schank, R.;Abelson, R.:" Scripts, plans, goals and understanding" Hillsdale, New York 1977

Small, G.W.; Weldon, L.J. :" Human factor studies of database query languages." Human Factors, 25, 253-263, 1983

Smith, P.;M. Barnes:" Files and databases." Addison Wesley 1987

Sparck Jones, K. :" Information Retrieval experiments" Butterworths, London 1981

Sparck-Jones, K.; Jackson, D.M.:" Current approaches to classification and clump-finding at the Cambridge Language Research Unit." Computer Journal 10, 1967, p.29-37

Sperberg-McQueen, C.M.; Burnard, L. (ed):" Guidelines for the encoding and interchange of machine readable texts" Chicago & Oxford 1990

Tague, J.M. :" The pragmatics of Information Retrieval experiments" in: Sparck-Jones(1981) :567:

Teskey, F.N.:" User models and world models for data, information and knowledge" Information processing and management, Vol. 25, no 1, 1989,pp. 7-14

Verharen, E. :" Hypercard en databases: een vooronderzoek naar de mogelijkheden van Hypercard." ITK memo, 1989

Waal, ? van de:" Some principles of a general iconographical classification." In: 'Actes du XVIIme congres internationale d'histoire d'art. Den haag 1955, pp.601-606

Waltz, D.:" Applications of the Connectionist machine." Thinking Machines Corporation 1986, Technical report 86-12

Winston, P.H.:" Artificial Intelligence" 2-th edit. Addison Wesley 1984

ITK: P.O. BOX 90153 5000 LE TILBURG THE NETHERLANDS